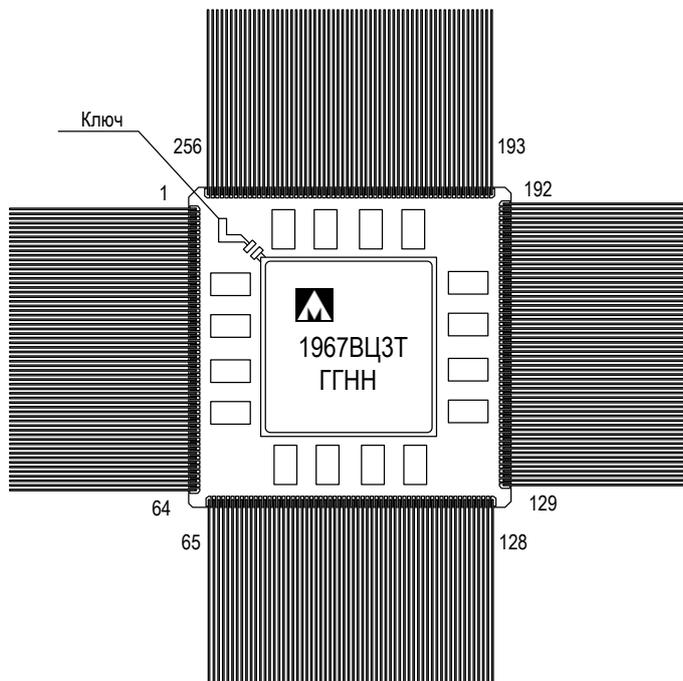




**Микросхема процессора цифровой обработки сигналов  
с ОЗУ 12 Мбит и тактовой частотой 300 МГц  
1967ВЦ3Т, К1967ВЦ3Т, К1967ВЦ3ТК**

**Основные параметры  
микросхемы**

- Два последовательных порта;
- Разрядность внешней шины данных 32 бит;
- Разрядность внешней шины адреса 22 бит;
- Напряжение питания ввода/вывода  $U_{CCIO}$  от 3,0 до 3,6 В;
- Динамический ток потребления схем ввода/вывода  $I_{OCC}$  не более 800 мА;
  
- Рабочий диапазон температур:



ГГ – год выпуска  
НН – неделя выпуска

Обозначение	Диапазон
1967ВЦ3Т	минус 60 – 105 °С
К1967ВЦ3Т	минус 60 – 105 °С
К1967ВЦ3ТК	0 – 70 °С

**Тип корпуса:**

- 256-ти выводной металлокерамический корпус 4244.256-3.

**Общее описание и области применения микросхемы**

Микросхемы интегральные 1967ВЦ3Т (далее – микросхемы) предназначены для использования в аппаратуре специального назначения в качестве процессора цифровой обработки сигналов с ОЗУ 12 Мбит и тактовой частотой 300 МГц.

## **Содержание**

1	Структурная блок-схема микросхемы.....	9
2	Условно-графическое обозначение .....	10
3	Описание выводов .....	11
4	Архитектура процессора ЦОС.....	23
4.1	Ядро процессора .....	27
4.1.1	Вычислительные модули.....	28
4.1.2	Целочисленное арифметико-логическое устройство (IALU).....	30
4.1.3	Устройство управления потоком команд.....	31
4.1.4	Управление ядром процессора.....	34
5	Память, регистры и шины.....	38
5.1	Пространство периферийных устройств SOC-шины .....	39
5.2	Пространство банка внешней памяти .....	39
5.3	Мультипроцессорное пространство .....	40
5.4	Внутреннее адресное пространство.....	40
5.5	Универсальные регистры .....	40
5.6	Группы регистров вычислительных модулей .....	42
5.7	Регистры вычислительного модуля без номера.....	45
5.7.1	Регистры статуса вычислительных модулей (XSTAT/YSTAT).....	45
5.7.2	Регистры АЛУ .....	46
5.7.3	Регистры умножителя .....	46
5.7.4	Регистры сдвигателя.....	47
5.7.5	Регистры блока CLU .....	47
5.8	Группы регистров целочисленного АЛУ .....	47
5.8.1	Регистры статусы целочисленного АЛУ (J31/JSTAT и K31/KSTAT).....	49
5.9	Группы регистров устройства управления.....	49
5.9.1	Регистр управления флагом (FLAGREG).....	50
5.9.2	Регистр управления SQCTL .....	50
5.9.3	Регистр управления (SQCTLST). Установка бит.....	51
5.9.4	Регистр управления (SQCTLCL). Сброс бит .....	51
5.9.5	Регистр статических флагов (SFREG).....	51
5.9.6	Регистр статуса устройства управления (SQSTAT).....	52
5.10	Группа регистров устройства защиты памяти .....	53
5.10.1	Регистры защиты (PUx) .....	54
5.10.2	Регистр управления (PU_CR).....	55
5.10.3	Регистр управления кэшированием данных .....	56
5.10.4	Регистр управления сквозной записью.....	56
5.10.5	Регистр управления кэшированием команд.....	56
5.11	Группы регистров отладки.....	56
5.11.1	Регистр управления точкой наблюдения (WPxCTL).....	57
5.11.2	Регистры состояния точки наблюдения (WPxSTAT) .....	59
5.11.3	Регистры указателя адреса точки наблюдения (WPxL/WPxH) .....	59
5.11.4	Регистр маски монитора производительности (PRFM).....	59
5.11.5	Регистр счетчика монитора производительности (PRFCNT).....	60
5.11.6	Регистры счетчика циклов (CCNTx).....	60
5.11.7	Регистры указателя и буфера трассировки (TRCBx/TRCBPTR).....	60
5.11.8	Регистр данных точки наблюдения 1 (WPDR) .....	61
5.11.9	Регистр маски точки наблюдения 1 (WPMR).....	62
6	Встроенный интерфейс (SOC-интерфейс).....	63
6.1	Транзакции SOC OFIFO.....	65
6.2	Транзакции SOC IFIFO .....	66

6.3	Транзакции SOC OBUF.....	66
6.4	Программирование SOC интерфейса .....	66
7	Таймеры.....	68
7.1	Регистр управления прерываниями (INTCTL) .....	69
7.2	Операции таймера.....	69
8	Порты общего назначения GPIO.....	71
8.1	Порты PA, PB, PC .....	71
8.2	Порты PxD и PxA.....	72
8.3	Регистр PX_ALT .....	73
8.4	Регистр FLAGREG .....	74
9	Прерывания.....	77
9.1	Группы регистров контроллера прерываний .....	78
9.1.1	Группы регистров векторов прерываний.....	78
9.1.2	Группа регистров управления контроллером прерываний .....	79
9.1.3	Регистр управления прерываниями (INTCTL).....	80
9.1.4	Регистры защелки прерываний (ILATL/ILATH).....	80
9.1.5	Регистры маскирования прерываний (IMASKL/IMASKH) .....	81
9.1.6	Регистры приоритетного маскирования прерываний (PMASKL/PMASKH).....	81
9.2	Операции с прерываниями .....	82
9.3	Программа обработки аппаратного прерывания.....	84
9.4	Особенности сохранения контекста процессора.....	87
9.5	Обработка программных прерываний (исключений).....	87
9.6	Обработка прерываний эмулятора.....	89
9.7	Источники прерываний процессора .....	90
9.7.1	Прерывания истечения таймера.....	91
9.7.2	Прерывания на обслуживание порта связи .....	91
9.7.3	Прерывания завершения работы каналов DMA .....	92
9.7.4	Прерывания от внешних источников .....	92
9.7.5	Векторное прерывание .....	92
9.7.6	Прерывание от захвата шины .....	92
9.7.7	Прерывание от аппаратной ошибки .....	92
9.7.8	Прерывания RTC.....	93
9.7.9	Прерывания от периферийных устройств .....	93
9.7.10	Программные исключения.....	93
10	Прямой доступ к памяти .....	95
10.1	Архитектура контроллера DMA.....	96
10.1.1	Регистры управления и статуса .....	99
10.1.2	Группа регистров TCB каналов 0-3 контроллер DMA.....	101
10.1.3	Группа регистров TCB каналов 4-7 контроллер DMA.....	102
10.1.4	Группа регистров TCB каналов 8-13.....	103
10.1.5	Группа регистров AutoDMA .....	103
10.2	Настройка DMA-передатч.....	104
10.2.1	Регистры блока управления передатчей (TCB).....	104
10.2.2	Регистр статуса DMA (DSTAT/DSTATC).....	109
10.2.3	Регистр управления DMA .....	110
10.3	Операции DMA контроллера.....	111
10.3.1	DMA управление каналами 4-11 .....	111
10.3.2	DMA управление каналами 0-3.....	112
10.3.3	DMA управление каналом 12 .....	112
10.4	DMA передатча.....	113
10.4.1	Адресация памяти.....	113
10.4.2	Приоритеты каналов DMA .....	113

10.4.3	Циклически присваиваемый приоритет .....	114
10.4.4	Приоритет контроллер DMA на внутренних шинах процессора ...	115
10.4.5	Цепочка DMA .....	116
10.4.6	Двухмерный DMA .....	117
10.4.7	Организация канала двухмерного DMA.....	118
10.4.8	Прерывания DMA .....	119
10.4.9	Запуск и окончание последовательностей DMA.....	120
10.5	Каналы DMA общего назначения .....	122
10.5.1	Передача из внешней памяти во внутреннюю память .....	123
10.5.2	Передача из внутренней памяти во внешнюю память .....	123
10.5.3	Передача из внешней памяти во внешнюю память .....	124
10.5.4	Передача из внутренней памяти во внутреннюю память .....	124
10.5.5	Передача из устройства ввода во внутреннюю или внешнюю память	125
10.5.6	Передача из внутренней или внешней памяти в устройство вывода	125
10.6	Каналы DMA с 4 по 11 для работы с устройствами .....	126
10.6.1	Передача из устройства во внутреннюю \ внешнюю память .....	127
10.6.2	Внутренняя/внешняя память – устройство-передатчик .....	127
10.6.3	Пересылка между портами связи .....	128
10.7	Канал 12 .....	128
11	Внешний порт .....	130
11.1	Внешняя память.....	130
11.1.1	Интерфейс внешней памяти .....	130
11.1.2	Команды контроллера SDRAM.....	157
11.1.3	Вспомогательные регистры интерфейса внешней шины .....	164
11.2	Поддержка мультипроцессорности .....	165
11.3	Хост-интерфейс .....	165
11.3.1	Регистры интерфейса .....	165
11.3.2	Аппаратная реализация интерфейса .....	166
12	Порты связи.....	173
12.1	Архитектура портов связи .....	173
12.2	Внешние выводы портов связи.....	175
12.3	Группа регистров портов связи.....	177
12.4	Прием и передача данных .....	178
12.5	Связь с DMA .....	178
12.6	Завершение блочной передачи .....	179
12.7	Прерывания портов связи .....	179
12.8	Инициализация после сброса и загрузка через порт связи.....	179
12.9	Протокол передачи данных порта связи.....	180
12.10	Задержки передачи через порт связи .....	183
12.11	Механизмы определения ошибок порта связи .....	184
12.11.1	Время ожидания передачи через порт связи.....	185
12.11.2	Время ожидания приемника .....	185
12.11.3	Ошибка верификации порта связи.....	185
12.11.4	Ошибка записи приема/передачи через порт связи .....	185
12.12	Регистр управления приемником порта связи (LRCTLx) .....	185
12.13	Регистр управления передатчиком порта связи (LTCTLx).....	187
12.14	Регистр состояния приемника порта связи (LRSTATx).....	187
12.15	Регистр состояния передатчика порта связи (LTSTATx) .....	188
13	Последовательный асинхронный интерфейс UART .....	189
13.1	Регистр данных UARTDR .....	190
13.2	Регистр состояния RXSTAT .....	190

13.3	Регистры управления скоростью обмена UBitRate .....	190
13.4	Регистр управления интерфейсом UCR .....	191
13.5	Регистр состояния интерфейса UFLAG .....	193
13.6	Регистры управления прерываниями UINT, UINTM .....	194
13.7	Программирование интерфейса UART .....	195
14	Последовательный синхронный интерфейс SPI .....	196
14.1	Регистр SPCR0.....	197
14.2	Регистр SPCR1.....	199
14.3	Регистр счета принимаемых данных RX_CNT .....	200
14.4	Регистр данных SPDR .....	201
14.5	Регистр состояния SPSR.....	201
14.6	Особенности работы в режиме “slave” .....	202
15	Контроллер видеокамеры.....	204
15.1	Регистры интерфейса.....	204
15.1.1	Регистр управления CR .....	204
15.1.2	Регистр состояния SR.....	204
15.1.3	Регистр данных DR .....	205
15.2	Режим приема «видеокамера» .....	205
15.3	Режим приема «ведущее устройство» .....	206
16	Контроллер внешней NAND флэш-памяти .....	208
16.1	Регистры управления контроллером.....	210
16.1.1	IO_CFG – регистр конфигурации временных параметров .....	210
16.1.2	WCT_CFG – регистр конфигурации времени ожидания .....	212
16.1.3	NAND_CFG – регистр конфигурации протокола обмена .....	213
16.1.4	WR_CFG и RD_CFG – регистры конфигурации .....	214
16.1.5	CR – регистр управления.....	215
16.1.6	DR – регистр данных.....	216
16.1.7	AR – регистр адреса .....	217
16.1.8	CNTR – счетчик количества слов .....	217
16.1.9	SR – регистр состояния .....	217
16.2	Чтение NAND флэш-памяти.....	218
16.3	Запись в NAND флэш-память .....	219
16.4	Типичные процедуры работы с NAND флэш-памятью фирмы Samsung .....	220
16.4.1	Сброс машины состояния флэш-памяти.....	220
16.4.2	Чтение ID .....	220
16.4.3	Чтение регистра состояния .....	220
16.4.4	Очистка (erase) блока флэш-памяти .....	221
16.4.5	Запись во флэш-память в последовательном режиме .....	221
16.4.6	Чтение из флэш-памяти в последовательном режиме .....	222
16.5	Рекомендации по организации работы с прерываниями.....	222
17	Контроллер LCD-панели.....	223
17.1	Регистр управления CTRL.....	226
17.2	Регистр состояния STATUS .....	228
17.3	Регистр управления сигналом fpline (HTIM).....	228
17.4	Регистр управления сигналом fframe (VTIM) .....	228
17.5	Регистр управления размером экрана (HVLEN).....	229
17.6	Регистр размера видео буфера (VSIZE) .....	229
17.7	Делитель для сигнала синхронизации панели (PXDV) .....	230
17.8	Управление горизонтальной активной областью панели (HDTIM) .....	230
17.9	Управление вертикальной активной областью панели (VDTIM).....	230
17.10	Управление дополнительной горизонтальной активной областью панели (HDxTIM) .....	231

17.11	Управление дополнительной вертикальной активной областью панели (VDxTIM) .....	231
17.12	Регистр FON .....	232
17.13	Регистр конфигурации панели (PANEL_CFG) .....	232
17.14	Регистр управления выходом ШИМ (PWM_CR) .....	232
17.15	Регистр управления сигналом GPIO_0, 1, 2, 3.....	233
17.16	Организация «спящего режима» .....	233
18	Интерфейс к аудио-кодеку AC97/I2S .....	235
18.1	Регистры интерфейса AC97/I2S .....	235
18.1.1	Регистр управления SICR0.....	235
18.1.2	Регистр управления SICR2.....	236
18.1.3	Регистр управления SICR3.....	236
18.1.4	Регистр состояния SISR .....	236
18.1.5	Регистр разрешения прерывания SIIER.....	238
18.1.6	Регистр запрещения прерывания SIIDR.....	239
18.1.7	Регистр адреса команды ACCAR.....	239
18.1.8	Регистр данных команды ACCDR .....	239
18.1.9	Регистр состояния адреса команды ACSAR .....	239
18.1.10	Регистр состояния данных команды ACSDR .....	240
18.1.11	Регистр данных GPIO ACGDR.....	240
18.1.12	Регистр состояния данных GPIO ACGSR.....	240
18.1.13	Регистр аудио данных SIADR.....	240
18.1.14	Регистр модемных данных SIMDR.....	241
18.2	Режимы работы.....	241
18.2.1	Режим работы AC97.....	241
18.2.2	Режим работы I2S .....	242
19	ARINC контроллер .....	247
19.1	Формат слова .....	248
19.2	Структурная схема канала приема.....	249
19.3	Структурная схема канала передачи .....	250
19.4	Описание регистров приемника.....	251
19.5	Регистр управления приемника RX_CR .....	252
19.5.1	Регистр состояния приемника STATUS.....	254
19.5.2	LABEL.....	255
19.5.3	DATA_R.....	255
19.6	Описание регистров передатчика.....	256
19.6.1	Регистр управления передатчиком CONTROL .....	256
19.6.2	Регистр состояния передатчика STATUS.....	257
19.6.3	DATA_T .....	258
20	Контроллер МКПД (MIL_STD).....	259
20.1	Режимы работы.....	260
20.1.1	Контроллер шины.....	260
20.1.2	Оконечное устройство .....	260
20.1.3	Монитор .....	261
20.2	Форматы сообщений.....	261
20.3	Формат слов .....	263
20.4	Структурная схема в режиме КШ.....	265
20.5	Структурная схема в режиме ОУ .....	266
20.6	Структурная схема в режиме М .....	267
20.7	Инициализация .....	267
20.8	Прием и передача в режиме ОУ .....	268
20.9	Прием и передача в режиме КШ.....	270
20.10	Прерывания.....	271

20.11	Описание регистров.....	271
20.11.1	Регистр управления CONTROL.....	272
20.11.2	Регистр состояния STATUS.....	273
20.11.3	Регистр ошибок ERROR .....	274
20.11.4	Регистр команды 1 CommandWord1 .....	275
20.11.5	Регистр команды 2 CommandWord2 .....	275
20.11.6	Слово данных команды управления ModeData .....	276
20.11.7	Ответное слово 1 StatusWord1.....	276
20.11.8	Ответное слово 2 StatusWord2.....	276
20.11.9	Регистр разрешения прерываний INTEN .....	277
20.11.10	Регистр декодирования сообщений MSG.....	278
20.11.11	Память принимаемых/передаваемых данных DATA.....	279
21	Модуль управления синхронизацией и энергопотреблением .....	280
21.1	Регистр конфигурации периферийных модулей CFG1 .....	280
21.2	Регистры управления внутренними PLL .....	281
21.2.1	Регистры CFG2, CFG3, CFG5.....	281
21.2.2	Регистры CFG4.....	283
21.3	Регистр состояния системы SYS_STS .....	284
21.4	Регистр управления синхронизацией периферийных устройств CFG8...285	
21.5	Переключение частоты процессора .....	285
21.6	Реализация «спящего режима» .....	286
22	Модуль цифрового коррелятора.....	289
22.1	Регистры ЦК .....	289
22.1.1	Регистр управления CR .....	290
22.1.2	Регистр состояния SR.....	291
22.1.3	Регистр количества LEN .....	291
22.1.4	Регистр конфигурации CFG.....	291
22.1.5	Регистр тестовых данных TST .....	292
22.1.6	Регистр запросов прерываний IRQ.....	292
22.2	Регистры канала .....	292
22.2.1	Регистры SCx_CNT и SCx_STEP .....	292
22.2.2	Регистры DZ_CNT и DZ_STEP .....	293
23	Порт JTAG и интерфейс отладки .....	295
23.1	Рабочие режимы .....	296
23.2	Ресурсы отладки .....	296
23.2.1	Специальные команды .....	296
23.2.2	Точки наблюдения.....	296
23.2.3	Буфер трассировки адреса команды (TBUF) .....	297
23.3	Мониторинг производительности .....	298
23.3.1	Регистр маски монитора производительности (PRFM) .....	298
23.3.2	Регистр счетчика монитора производительности (PRFCNT).....	299
23.3.3	Регистры счетчика циклов (CCNTx).....	299
23.3.4	Регистр данных точки наблюдения 1 (WPDR) .....	300
23.3.5	Регистр маски точки наблюдения 1 (WPMR).....	300
23.4	JTAG интерфейс .....	300
23.4.1	Выводы JTAG порта.....	300
23.4.2	Группа регистров эмулятора JTAG .....	301
23.4.3	Регистр команды JTAG .....	302
23.4.4	Регистры данных.....	303
24	Начальный старт процессора .....	305
24.1	Загрузка из внешней NAND флэш памяти .....	305
24.2	Загрузка EPROM .....	306
24.3	Загрузка с использованием порта связи .....	307

24.4	Старт по запросу прерывания.....	309
24.5	Загрузка из внешней последовательной флэш памяти .....	310
25	Часы реального времени RTC .....	312
25.1	Формирователь «тик-импульсов» .....	312
25.2	Формирователь импульсов секунд .....	313
25.3	Счетчик секунд RTC_CNT. ....	313
25.3.1	Регистр сравнения RTC_MR .....	313
25.3.2	Регистр управления RTC_CR.....	313
25.3.3	Регистр счетчика сторожевого таймера WDT_CNT .....	314
25.3.4	Регистр занятости интерфейса часов RTC_BUSY .....	314
25.4	Ограничения модуля RTC .....	315
25.5	Состояние модуля RTC после сброса.....	315
26	Предельно-допустимые режимы.....	316
27	Электрические параметры .....	317
28	Габаритный чертеж микросхемы .....	318
29	Информация для заказа .....	319
	Лист регистрации изменений .....	320

# 1 Структурная блок-схема микросхемы

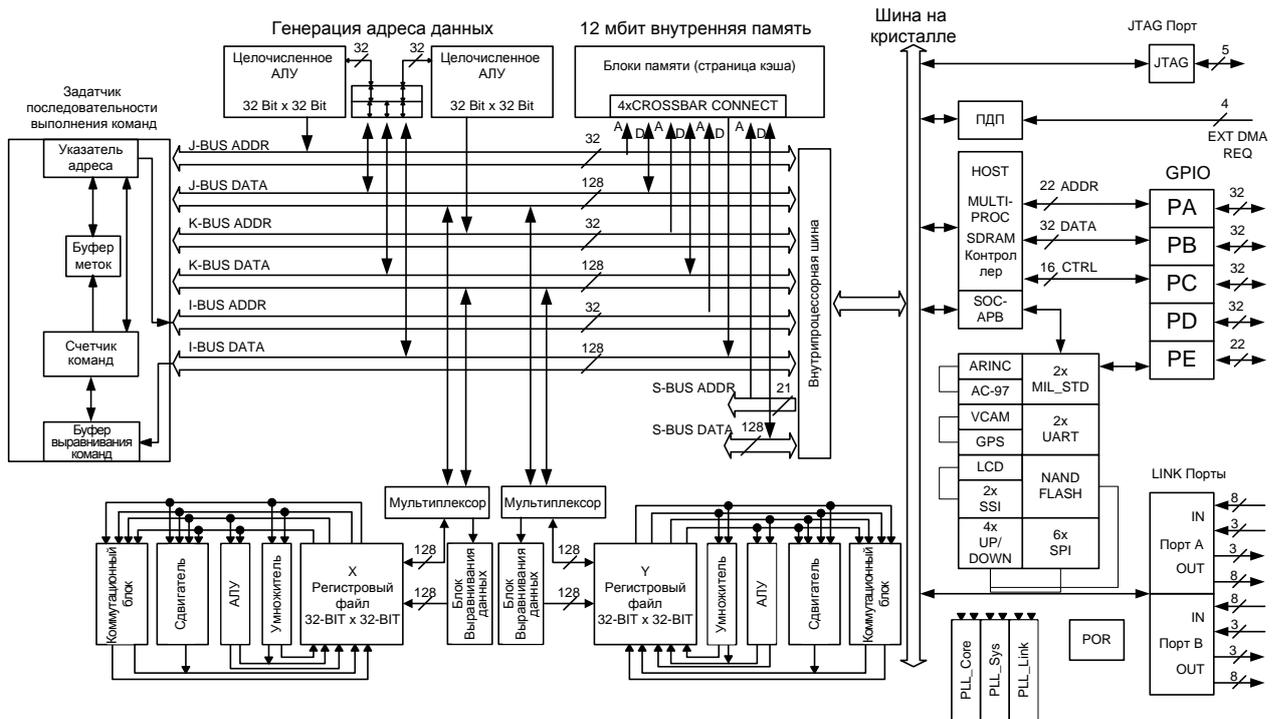


Рисунок 1 – Структурная блок-схема микросхемы

## 2 Условно-графическое обозначение

		1987ВЦЗТ		
13-26, 18-33, 35-40, 42-47	Порт А PA[0]-PA[31]		Порт С PC[0]-PC[31]	82-89, 91-102, 104-108, 197, 207- 212, 219
48-74, 76-80	Порт В PB[31]-PB[0]		Порт Е PE[31]-PE[0]	109-123, 125-131
			Порт D PD[31]-PD[0]	132-136, 138-147, 149-165
187-196, 199-204	LODATI[0]- LODATI[7]		LODATO[7]- LODATO[0]	167-172, 174-177, 179-184
205	LOCLKINN		LOCLKOUTN	185
206	LOCLKINP		LOCLKOUTP	186
236, 237, 239-252	L1DATO[7]- L1DATO[0]		L1DATO[7]- L1DATO[0]	214-217, 220-223, 225-232
253	L1CLKINN		L1CLKOUTN	234
254	L1CLKINP		L1CLKOUTP	235
255	nTRST			
1	TCK			
2	TMS			
3	TDI			
4	TDO			
6	JG_MX		nEMU	5
7	nRST_IN			
12	XTI		XTO	11
34, 81, 103, 124, 148, 178, 213, 233	VDD_CORE			8, 10, 41, 75, 97, 137, 166, 198, 218, 238, 256
27, 90, 173, 224	VDD_IO		VSS	
9	VDDA			

Рисунок 2 – Условно-графическое обозначение

### 3 Описание выводов

Таблица 1 – Описание выводов процессора

Номер вывода	Обозначение вывода	Дополнительное назначение вывода	Тип вывода	Функциональное назначение
1	TCK	–	I	Интерфейс JTAG. Синхросигнал (резистор доопределения до 1)
2	TMS	–	I	Интерфейс JTAG Выбор режима Синхросигнал (резистор доопределения до 1)
3	TDI	–	I	Интерфейс JTAG Вход данных Синхросигнал (резистор доопределения до 1)
4	TDO	–	O	Интерфейс JTAG Выход данных
5	nEMU	–	O	Индикатор режима отладки
6	JG_MX	–	I	Выбор блока JTAG (резистор доопределения до 0)
7	nRST_IN	–	I	Общий сброс (активный низкий уровень)
8	VSS	–	GND	Общий
9	VDDA	–	PWR	Питание аналоговых блоков
10	VSS	–	GND	Общий
11	XTO	–	–	Вывод кварцевого генератора
12	XTI	–	–	Вывод кварцевого генератора
13	PA[0]/	U1_TXD	IO	Порт А общего назначения / Интерфейс UART1. Выход передатчика (резистор доопределения до 1)
14	PA[1]	U1_RXD	IO	Порт А общего назначения / Интерфейс UART1. Вход приемника (резистор доопределения до 1)
15	PA[2]	U2_TXD	IO	Порт А общего назначения / Интерфейс UART2. Выход передатчика (резистор доопределения до 1)
16	PA[3]	U2_RXD	IO	Порт А общего назначения / Интерфейс UART2. Вход приемника (резистор доопределения до 1)
17	PA[4]	SPI_CLK	IO	Порт А общего назначения / Интерфейс SPI. Синхросигнал (резистор доопределения до 1)
18	PA[5]	SPI_DO	IO	Порт А общего назначения/ Интерфейс SPI. Выход данных SPI (резистор доопределения до 1)
19	PA[6]	SPI_DI	IO	Порт А общего назначения/ Интерфейс SPI. Вход данных 0 SPI (резистор доопределения до 1)
20	PA[7]	SPI_CS[0]	O	Порт А общего назначения/ Интерфейс SPI. Выбор устройства 0 (резистор доопределения до 1)
21	PA[8]	SPI_CS[1]	O	Порт А общего назначения/ Интерфейс SPI. Выбор устройства 0 (резистор доопределения до 1)
22	PA[9]	SPI_CS[2]	O	Порт А общего назначения/ Интерфейс SPI. Выбор устройства 0 (резистор доопределения до 1)

**Спецификация 1967ВЦ3Т, К1967ВЦ3Т, К1967ВЦ3ТК**

Номер вывода	Обозначение вывода	Дополнительное назначение вывода	Тип вывода	Функциональное назначение
23	PA[10]	SPI_CS[3]	O	Порт А общего назначения/ Интерфейс SPI. Выбор устройства 0 (резистор доопределения до 1)
24	PA[11]	SPI_CS[4]	O	Порт А общего назначения/ Интерфейс SPI. Выбор устройства 0 (резистор доопределения до 1)
25	PA[12]	SPI_CS[5]	O	Порт А общего назначения/ Интерфейс SPI. Выбор устройства 0 (резистор доопределения до 1)
26	PA[13]	SSI1_TCLK	IO	Порт А общего назначения/ Интерфейс SSI(1). Синхросигнал передатчика (резистор доопределения до 1)
27	VDD_IO	–	PWR	Питание площадок ввода/вывода
28	PA[14]	SSI1_TFS	IO	Порт А общего назначения/ Интерфейс SSI(1). Строб начала кадра или выбор канала (резистор доопределения до 1)
29	PA[15]	SSI1_TXD	IO	Порт А общего назначения/ Интерфейс SSI(1). Данные передатчика (резистор доопределения до 1)
30	PA[16]	SSI1_RCLK	IO	Порт А общего назначения/ Интерфейс SSI(1). Синхросигнал приемника (резистор доопределения до 1)
31	PA[17]	SSI1_RFS	IO	Порт А общего назначения/ Интерфейс SSI(1). Строб начала кадра или выбор канала (резистор доопределения до 1)
32	PA[18]	SSI1_RXD	IO	Порт А общего назначения/ Интерфейс SSI(1). Данные приемника (резистор доопределения до 1)
33	PA[19]	SSI2_TCLK	IO	Порт А общего назначения/ Интерфейс SSI(2). Синхросигнал передатчика (резистор доопределения до 1)
34	VDD_CORE	–	PWR	Питание ядра
35	PA[20]	SSI2_TFS	IO	Порт А общего назначения/ Интерфейс SSI(2). Строб начала кадра или выбор канала (резистор доопределения до 1)
36	PA[21]	SSI2_TXD	IO	Порт А общего назначения/ Интерфейс SSI(2). Данные передатчика (резистор доопределения до 1)
37	PA[22]	SSI2_RCLK	IO	Порт А общего назначения/ Интерфейс SSI(2). Синхросигнал приемника (резистор доопределения до 1)
38	PA[23]	SSI2_RFS	IO	Порт А общего назначения/ Интерфейс SSI(2). Строб начала кадра или выбор канала (резистор доопределения до 1)
39	PA[24]	SSI2_RXD	IO	Порт А общего назначения/ Интерфейс SSI(2). Данные приемника (резистор доопределения до 1)
40	PA[25]	VC_CLK	IO	Порт А общего назначения/ Видеоинтерфейс. Синхросигнал (резистор доопределения до 1)
41	VSS	–	GND	Общий

**Спецификация 1967ВЦЗТ, К1967ВЦЗТ, К1967ВЦЗТК**

Номер вывода	Обозначение вывода	Дополнительное назначение вывода	Тип вывода	Функциональное назначение
42	PA[26]	VC_VSYNC	IO	Порт А общего назначения/ Видеоинтерфейс. Вертикальная развертка / Интерфейс MIL-STD(1) (резистор доопределения до 1)
43	PA[27]	VC_HSYNC	IO	Порт А общего назначения/ Видеоинтерфейс. Горизонтальная развертка / Интерфейс MIL-STD(1) (резистор доопределения до 1)
44	PA[28]	VC_DATA[0]	IO	Порт А общего назначения/ Видеоинтерфейс. Бит 0 данных / Интерфейс MIL-STD(1) (резистор доопределения до 1)
45	PA[29]	VC_DATA[1]	IO	Порт А общего назначения/ Видеоинтерфейс. Бит 1 данных (резистор доопределения до 1)
46	PA[30]	VC_DATA[2]	IO	Порт А общего назначения/ Видеоинтерфейс. Бит 2 данных / Интерфейс MIL-STD(1) (резистор доопределения до 1)
47	PA[31]	VC_DATA[3]	IO	Порт А общего назначения/ Видеоинтерфейс. Бит 3 данных / Интерфейс MIL-STD(1) (резистор доопределения до 1)
48	PB[0]	VC_DATA[4]	I	Порт В общего назначения/ Видеоинтерфейс. Бит 4 данных / Интерфейс MIL-STD(1) (резистор доопределения до 1)
49	PB[1]	VC_DATA[5]	I	Порт В общего назначения / Видеоинтерфейс. Бит 5 данных / Интерфейс MIL-STD(1) (резистор доопределения до 1)
50	PB[2]	VC_DATA[6]	I	Порт В общего назначения / Видеоинтерфейс. Бит 6 данных / Интерфейс MIL-STD(1) (резистор доопределения до 1)
51	PB[3]	VC_DATA[7]	I	Порт В общего назначения / Видеоинтерфейс. Бит 7 данных / Интерфейс MIL-STD(1) (резистор доопределения до 1)
52	PB[4]	LC_B[0]	IO	Порт В общего назначения / ЖКИ. Бит 0 синего цвета / Контроллер ARINC. Вход / Запрос канала 0 контроллера DMA (резистор доопределения до 1)
53	PB[5]	LC_B[1]	IO	Порт В общего назначения / ЖКИ. Бит 1 синего цвета / Контроллер ARINC. Вход (резистор доопределения до 1)
54	PB[6]	LC_B[2]	IO	Порт В общего назначения / ЖКИ. Бит 2 синего цвета / Контроллер ARINC. Вход (резистор доопределения до 1)
55	PB[7]	LC_B[3]	IO	Порт В общего назначения / ЖКИ. Бит 3 синего цвета / Контроллер ARINC. Вход (резистор доопределения до 1)
56	PB[8]	LC_B[4]	IO	Порт В общего назначения / ЖКИ. Бит 4 синего цвета / Контроллер ARINC. Вход (резистор доопределения до 1)
57	PB[9]	LC_B[5]	IO	Порт В общего назначения / ЖКИ. Бит 5 синего цвета / Контроллер ARINC. Вход (резистор доопределения до 1)
58	PB[10]	LC_G[0]	IO	Порт В общего назначения / ЖКИ. Бит 0 зеленого цвета / Контроллер ARINC. Вход (резистор доопределения до 1)
59	PB[11]	LC_G[1]	IO	Порт В общего назначения / ЖКИ. Бит 1 зеленого цвета / Контроллер ARINC. Вход (резистор доопределения до 1)

**Спецификация 1967ВЦ3Т, К1967ВЦ3Т, К1967ВЦ3ТК**

Номер вывода	Обозначение вывода	Дополнительное назначение вывода	Тип вывода	Функциональное назначение
60	PВ[12]	LC_G[2]	IO	Порт В общего назначения / ЖКИ. Бит 2 зеленого цвета / Контроллер ARINC. Вход / Интерфейс MIL-STD(1) (резистор доопределения до 1)
61	PВ[13]	LC_G[3]	IO	Порт В общего назначения / ЖКИ. Бит 3 зеленого цвета / Контроллер ARINC. Вход / Интерфейс MIL-STD(1) (резистор доопределения до 1)
62	PВ[14]	LC_G[4]	IO	Порт В общего назначения / ЖКИ. Бит 4 зеленого цвета / Контроллер ARINC. Вход / Интерфейс MIL-STD(1) (резистор доопределения до 1)
63	PВ[15]	LC_G[5]	IO	Порт В общего назначения / ЖКИ. Бит 5 зеленого цвета / Контроллер ARINC. Вход / Интерфейс MIL-STD(1) (резистор доопределения до 1)
64	PВ[16]	LC_R[0]	IO	Порт В общего назначения / ЖКИ. Бит 0 красного цвета / Контроллер ARINC. Вход / Запрос канала 1 контроллера DMA (резистор доопределения до 1)
65	PВ[17]	LC_R[1]	IO	Порт В общего назначения / ЖКИ. Бит 1 красного цвета / Контроллер ARINC. Вход (резистор доопределения до 1)
66	PВ[18]	LC_R[2]	IO	Порт В общего назначения / ЖКИ. Бит 2 красного цвета / Контроллер ARINC / Интерфейс MIL-STD(1) (резистор доопределения до 1)
67	PВ[19]	LC_R[3]	IO	Порт В общего назначения / ЖКИ. Бит 3 красного цвета / Контроллер ARINC / Интерфейс MIL-STD(1) (резистор доопределения до 1)
68	PВ[20]	LC_R[4]	IO	Порт В общего назначения / ЖКИ. Бит 3 красного цвета / Контроллер ARINC. Выход (резистор доопределения до 1)
69	PВ[21]	LC_R[5]	IO	Порт В общего назначения / ЖКИ. Бит 3 красного цвета / Контроллер ARINC. Выход (резистор доопределения до 1)
70	PВ[22]	LC_T[0]	IO	Порт В общего назначения / ЖКИ. Дополнительный выход 0 управления / Вход прерывания 0 (резистор доопределения до 1)
71	PВ[23]	LC_T[1]	IO	Порт В общего назначения / ЖКИ. Дополнительный выход 1 управления / Вход прерывания 1 (резистор доопределения до 1)
72	PВ[24]	LC_T[2]	IO	Порт В общего назначения / ЖКИ. Дополнительный выход 2 управления / Вход прерывания 2 (резистор доопределения до 1)
73	PВ[25]	LC_T[3]	IO	Порт В общего назначения / ЖКИ. Дополнительный выход 3 управления / Вход прерывания 3 (резистор доопределения до 1)
74	PВ[26]	LC_PWM	IO	Порт В общего назначения / ЖКИ. Выход ШИМ управления яркостью / Контроллер ARINC. Выход / Запрос канала 2 контроллера DMA (резистор доопределения до 1)
75	VSS		GND	Общий

**Спецификация 1967ВЦ3Т, К1967ВЦ3Т, К1967ВЦ3ТК**

Номер вывода	Обозначение вывода	Дополнительное назначение вывода	Тип вывода	Функциональное назначение
76	PB[27]	LC_DRDY	IO	Порт В общего назначения / ЖКИ. Флаг готовности данных / Контроллер ARINC. Выход (резистор доопределения до 1)
77	PB[28]	LC_VSYNC	IO	Порт В общего назначения / ЖКИ. Вертикальная синхронизация / Контроллер ARINC. Выход (резистор доопределения до 1)
78	PB[29]	LC_HSYNC	IO	Порт В общего назначения / ЖКИ. Горизонтальная синхронизация / Контроллер ARINC. Выход (резистор доопределения до 1)
79	PB[30]	LC_CLK	IO	Порт В общего назначения / ЖКИ. Синхросигнал / Контроллер ARINC. Выход (резистор доопределения до 1)
80	PB[31]	nDMAR3	IO	Порт В общего назначения / Запрос канала 3 контроллера DMA / Контроллер ARINC. Выход (резистор доопределения до 1)
81	VDD_CORE	–	PWR	Питание ядра
82	PC[0]	FLAG[0]	IO	Порт С общего назначения / Флаг 0 (резистор доопределения до 1)
83	PC[1]	FLAG[1]	IO	Порт С общего назначения / Флаг 1 (резистор доопределения до 1)
84	PC[2]	FLAG[2]	IO	Порт С общего назначения / Флаг 2 (резистор доопределения до 1)
85	PC[3]	FLAG[3]	IO	Порт С общего назначения / Флаг 3 (резистор доопределения до 1)
86	PC[4]	BOOT[0]	IO	Порт С общего назначения / Выбор режима загрузки (резистор доопределения до 1)
87	PC[5]	BOOT[1]	I	Порт С общего назначения / Выбор режима загрузки (резистор доопределения до 1)
88	PC[6]	BOOT[2]	IO	Порт С общего назначения / Выбор режима загрузки (резистор доопределения до 1)
89	PC[7]	SCLK	O	Внешний интерфейс. Синхронсигнал
90	VDD_IO	–	PWR	Питание контактных площадок
91	PC[8]	SD_CKE	IO	Порт С общего назначения / Контроллер SDRAM. Разрешение синхросигнала (резистор доопределения до 1)
92	PC[9]	MSSD[0]	IO	Порт С общего назначения / Контроллер SDRAM. Выбор модуля 0 (резистор доопределения до 1)
93	PC[10]	MSSD[1]	IO	Порт С общего назначения / Контроллер SDRAM. Выбор модуля 1 (резистор доопределения до 1)
94	PC[11]	MSSD[2]	IO	Порт С общего назначения / Контроллер SDRAM. Выбор модуля 2 (резистор доопределения до 1)
95	PC[12]	MSSD[3]	IO	Порт С общего назначения / Контроллер SDRAM. Выбор модуля 3 (резистор доопределения до 1)
96	PC[13]	SD_nCAS	IO	Порт С общего назначения / Контроллер SDRAM. Выбор колонки (резистор доопределения до 1)

**Спецификация 1967ВЦ3Т, К1967ВЦ3Т, К1967ВЦ3ТК**

Номер вывода	Обозначение вывода	Дополнительное назначение вывода	Тип вывода	Функциональное назначение
97	VSS	–	GND	Общий
98	PC[14]	SD_nRAS	IO	Порт С общего назначения / Контроллер SDRAM. Выбор строки (резистор доопределения до 1)
99	PC[15]	SD_nWE	IO	Порт С общего назначения / Контроллер SDRAM. Разрешение записи (резистор доопределения до 1)
100	PC[16]	SD_DQM	IO	Порт С общего назначения / Контроллер SDRAM. Маска (резистор доопределения до 1)
101	PC[17]	SD_A10	IO	Порт С общего назначения / Контроллер SDRAM. Бит 10 адреса (резистор доопределения до 1)
102	PC[18]	nMS[1]	O	Порт С общего назначения / Контроллер SDRAM. Выбор статической памяти тип 1 (резистор доопределения до 1)
103	VDD_CORE	–	PWR	Питание ядра
104	PC[19]	nMS[0]	O	Порт С общего назначения / Контроллер SRAM. Выбор статической памяти тип 0 (резистор доопределения до 1)
105	PC[20]	nBMS	O	Порт С общего назначения / Контроллер EEPROM. Выбор модуля (резистор доопределения до 1)
106	PC[21]	nRD	IO	Порт С общего назначения / Контроллер SRAM. Строб чтения (резистор доопределения до 1)
107	PC[22]	nWR	IO	Порт С общего назначения / Контроллер SRAM. Выбор статической памяти тип 0 (резистор доопределения до 1)
108	PC[23]	ACK	IO	Порт С общего назначения / Контроллер SRAM. Флаг готовности статической памяти (резистор доопределения до 1)
109	PE[21]	ADDR[21]	O	Шина адреса внешнего интерфейса. Бит 21
110	PE[20]	ADDR[20]	O	Шина адреса внешнего интерфейса. Бит 20
111	PE[19]	ADDR[19]	O	Шина адреса внешнего интерфейса. Бит 19
112	PE[18]	ADDR[18]	O	Шина адреса внешнего интерфейса. Бит 18
113	PE[17]	ADDR[17]	O	Шина адреса внешнего интерфейса. Бит 17
114	PE[16]	ADDR[16]	O	Шина адреса внешнего интерфейса. Бит 16
115	PE[15]	ADDR[15]	O	Шина адреса внешнего интерфейса. Бит 15
116	PE[14]	ADDR[14]	O	Шина адреса внешнего интерфейса. Бит 14
117	PE[13]	ADDR[13]	O	Шина адреса внешнего интерфейса. Бит 13
118	PE[12]	ADDR[12]	O	Шина адреса внешнего интерфейса. Бит 12
119	PE[11]	ADDR[11]	O	Шина адреса внешнего интерфейса. Бит 11
120	PE[10]	ADDR[10]	O	Шина адреса внешнего интерфейса. Бит 10
121	PE[9]	ADDR[9]	O	Шина адреса внешнего интерфейса. Бит 9
122	PE[8]	ADDR[8]	O	Шина адреса внешнего интерфейса. Бит 8
123	PE[7]	ADDR[7]	O	Шина адреса внешнего интерфейса. Бит 7
124	VDD_CORE	–	PWR	Питание ядра
125	PE[6]	ADDR[6]	O	Шина адреса внешнего интерфейса. Бит 6
126	PE[5]	ADDR[5]	O	Шина адреса внешнего интерфейса. Бит 5
127	PE[4]	ADDR[4]	O	Шина адреса внешнего интерфейса. Бит 4

**Спецификация 1967ВЦ3Т, К1967ВЦ3Т, К1967ВЦ3ТК**

Номер вывода	Обозначение вывода	Дополнительное назначение вывода	Тип вывода	Функциональное назначение
128	PE[3]	ADDR[3]	О	Шина адреса внешнего интерфейса. Бит 3
129	PE[2]	ADDR[2]	О	Шина адреса внешнего интерфейса. Бит 2
130	PE[1]	ADDR[1]	О	Шина адреса внешнего интерфейса. Бит 1
131	PE[0]	ADDR[0]	О	Шина адреса внешнего интерфейса. Бит 0
132	PD[31]	DATA[31]	IO	Шина данных внешнего интерфейса. Бит 31 / Контроллер NAND FLASH. Бит готовности / Интерфейс MIL-STD(2) (резистор доопределения до 1)
133	PD[30]	DATA[30]	IO	Шина данных внешнего интерфейса. Бит 30 / Контроллер NAND FLASH. Выбор модуля 2 / Интерфейс MIL-STD(2) (резистор доопределения до 1)
134	PD[29]	DATA[29]	IO	Шина данных внешнего интерфейса. Бит 29 / Контроллер NAND FLASH. Выбор модуля 1 / Интерфейс MIL-STD(2) (резистор доопределения до 1)
135	PD[28]	DATA[28]	IO	Шина данных внешнего интерфейса. Бит 28 / Контроллер NAND FLASH. Выбор модуля 0 / Интерфейс MIL-STD(2) (резистор доопределения до 1)
136	PD[27]	DATA[27]	IO	Шина данных внешнего интерфейса. Бит 27 / Контроллер NAND FLASH. Строб записи / Интерфейс MIL-STD(2) (резистор доопределения до 1)
137	VSS	—	GND	Общий
138	PD[26]	DATA[26]	IO	Шина данных внешнего интерфейса. Бит 26 / Контроллер NAND FLASH. Строб чтения / Интерфейс MIL-STD(2) (резистор доопределения до 1)
139	PD[25]	DATA[25]	IO	Шина данных внешнего интерфейса. Бит 25 / Контроллер NAND FLASH. Строб адреса / Интерфейс MIL-STD(2) (резистор доопределения до 1)
140	PD[24]	DATA[24]	IO	Шина данных внешнего интерфейса. Бит 24 / Контроллер NAND FLASH. Строб команды / Интерфейс MIL-STD(2) (резистор доопределения до 1)
141	PD[23]	DATA[23]	IO	Шина данных внешнего интерфейса. Бит 23 / Контроллер NAND FLASH. Шина данных, бит 7 / Интерфейс MIL-STD(2) (резистор доопределения до 1)
142	PD[22]	DATA[22]	IO	Шина данных внешнего интерфейса. Бит 22 / Контроллер NAND FLASH. Шина данных, бит 6 / Интерфейс MIL-STD(2) (резистор доопределения до 1)
143	PD[21]	DATA[21]	IO	Шина данных внешнего интерфейса. Бит 21 / - Контроллер NAND FLASH. Шина данных, бит 5 / Интерфейс MIL-STD(2) (резистор доопределения до 1)
144	PD[20]	DATA[20]	IO	Шина данных внешнего интерфейса. Бит 20 / Контроллер NAND FLASH. Шина данных, бит 4 / Интерфейс MIL-STD(2) (резистор доопределения до 1)

**Спецификация 1967ВЦ3Т, К1967ВЦ3Т, К1967ВЦ3ТК**

Номер вывода	Обозначение вывода	Дополнительное назначение вывода	Тип вывода	Функциональное назначение
145	PD[19]	DATA[19]	IO	Шина данных внешнего интерфейса. Бит 19 / Контроллер NAND FLASH. Шина данных, бит 3 / Интерфейс MIL-STD(2) (резистор доопределения до 1)
146	PD[18]	DATA[18]	IO	Шина данных внешнего интерфейса. Бит 18 / Контроллер NAND FLASH. Шина данных, бит 2 / Интерфейс MIL-STD(2) (резистор доопределения до 1)
147	PD[17]	DATA[17]	IO	Шина данных внешнего интерфейса. Бит 17 / Контроллер NAND FLASH. Шина данных, бит 1 / Интерфейс MIL-STD(2) (резистор доопределения до 1)
148	VDD_CORE	–	PWR	Питание ядра
149	PD[16]	DATA[16]	IO	Шина данных внешнего интерфейса. Бит 16 / Контроллер NAND FLASH. Шина данных, бит 0 / Интерфейс MIL-STD(2) (резистор доопределения до 1)
150	PD[15]	DATA[15]	IO	Шина данных внешнего интерфейса. Бит 15 (резистор доопределения до 1)
151	PD[14]	DATA[14]	IO	Шина данных внешнего интерфейса. Бит 14 (резистор доопределения до 1)
152	PD[13]	DATA[13]	IO	Шина данных внешнего интерфейса. Бит 13 (резистор доопределения до 1)
153	PD[12]	DATA[12]	IO	Шина данных внешнего интерфейса. Бит 12 (резистор доопределения до 1)
154	PD[11]	DATA[11]	IO	Шина данных внешнего интерфейса. Бит 11 (резистор доопределения до 1)
155	PD[10]	DATA[10]	IO	Шина данных внешнего интерфейса. Бит 10 (резистор доопределения до 1)
156	PD[9]	DATA[9]	IO	Шина данных внешнего интерфейса. Бит 9 (резистор доопределения до 1)
157	PD[8]	DATA[8]	IO	Шина данных внешнего интерфейса. Бит 8 (резистор доопределения до 1)
158	PD[7]	DATA[7]	IO	Шина данных внешнего интерфейса. Бит 7 (резистор доопределения до 1)
159	PD[6]	DATA[6]	IO	Шина данных внешнего интерфейса. Бит 6 (резистор доопределения до 1)
160	PD[5]	DATA[5]	IO	Шина данных внешнего интерфейса. Бит 5 (резистор доопределения до 1)
161	PD[4]	DATA[4]	IO	Шина данных внешнего интерфейса. Бит 4 (резистор доопределения до 1)
162	PD[3]	DATA[3]	IO	Шина данных внешнего интерфейса. Бит 3 (резистор доопределения до 1)
163	PD[2]	DATA[2]	IO	Шина данных внешнего интерфейса. Бит 2 (резистор доопределения до 1)
164	PD[1]	DATA[1]	IO	Шина данных внешнего интерфейса. Бит 1 (резистор доопределения до 1)
165	PD[0]	DATA[0]	IO	Шина данных внешнего интерфейса. Бит 0 (резистор доопределения до 1)
166	VSS	–	GND	Общий
167	L0DATON[0]	–	O	LINK интерфейс 0. Выход данных (отрицательный), бит 0

**Спецификация 1967ВЦ3Т, К1967ВЦ3Т, К1967ВЦ3ТК**

Номер вывода	Обозначение вывода	Дополнительное назначение вывода	Тип вывода	Функциональное назначение
168	L0DATOP[0]	–	О	LINK интерфейс 0. Выход данных (положительный), бит 0
169	L0DATON[1]	–	О	LINK интерфейс 0. Выход данных (отрицательный), бит 1
170	L0DATOP[1]	–	О	LINK интерфейс 0. Выход данных (положительный), бит 1
171	L0DATON[2]	–	О	LINK интерфейс 0. Выход данных (отрицательный), бит 2
172	L0DATOP[2]	–	О	LINK интерфейс 0. Выход данных (положительный), бит 2
173	VDD_IO	–	PWR	Питание контактных площадок
174	L0DATON[3]	–	О	LINK интерфейс 0. Выход данных (отрицательный), бит 3
175	L0DATOP[3]	–	О	LINK интерфейс 0. Выход данных (положительный), бит 3
176	L0DATON[4]	–	О	LINK интерфейс 0. Выход данных (отрицательный), бит 4
177	L0DATOP[4]	–	О	LINK интерфейс 0. Выход данных (положительный), бит 4
178	VDD_CORE	–	PWR	Питание ядра
179	L0DATON[5]	–	О	LINK интерфейс 0. Выход данных (отрицательный), бит 5
180	L0DATOP[5]	–	О	LINK интерфейс 0. Выход данных (положительный), бит 5
181	L0DATON[6]	–	О	LINK интерфейс 0. Выход данных (отрицательный), бит 6
182	L0DATOP[6]	–	О	LINK интерфейс 0. Выход данных (положительный), бит 6
183	L0DATON[7]	–	О	LINK интерфейс 0. Выход данных (отрицательный), бит 7
184	L0DATOP[7]	–	О	LINK интерфейс 0. Выход данных (положительный), бит 7
185	L0CLKOUTN	–	О	LINK интерфейс 0. Выход синхросигнала (отрицательный)
186	L0CLKOUTP	–	О	LINK интерфейс 0. Выход синхросигнала (положительный)
187	L0DATIN[0]	–	I	LINK интерфейс 0. Вход данных (отрицательный), бит 0
188	L0DATIP[0]	–	I	LINK интерфейс 0. Вход данных (положительный), бит 0

**Спецификация 1967ВЦ3Т, К1967ВЦ3Т, К1967ВЦ3ТК**

Номер вывода	Обозначение вывода	Дополнительное назначение вывода	Тип вывода	Функциональное назначение
189	L0DATIN[1]	–	I	LINK интерфейс 0. Вход данных (отрицательный), бит 1
190	L0DATIP[1]	–	I	LINK интерфейс 0. Вход данных (положительный), бит 1
191	L0DATIN[2]	–	I	LINK интерфейс 0. Вход данных (отрицательный), бит 2
192	L0DATIP[2]	–	I	LINK интерфейс 0. Вход данных (положительный), бит 2
193	L0DATIN[3]	–	I	LINK интерфейс 0. Вход данных (отрицательный), бит 3
194	L0DATIP[3]	–	I	LINK интерфейс 0. Вход данных (положительный), бит 3
195	L0DATIN[4]	–	I	LINK интерфейс 0. Вход данных (отрицательный), бит 4
196	L0DATIP[4]	–	I	LINK интерфейс 0. Вход данных (положительный), бит 4
197	PC[24]	L0ACKO	IO	LINK интерфейс 0. Выход сигнала готовности
198	VSS	–	GND	Общий
199	L0DATIN[5]	–	I	LINK интерфейс 0. Вход данных (отрицательный), бит 5
200	L0DATIP[5]	–	I	LINK интерфейс 0. Вход данных (положительный), бит 5
201	L0DATIN[6]	–	I	LINK интерфейс 0. Вход данных (отрицательный), бит 6
202	L0DATIP[6]	–	I	LINK интерфейс 0. Вход данных (положительный), бит 6
203	L0DATIN[7]	–	I	LINK интерфейс 0. Вход данных (отрицательный), бит 7
204	L0DATIP[7]	–	I	LINK интерфейс 0. Вход данных (положительный), бит 7
205	L0CLKINN	–	I	LINK интерфейс 0. Вход синхросигнала (отрицательный)
206	L0CLKINP	–	I	LINK интерфейс 0. Вход синхросигнала (положительный)
207	PC[25]	L0ACKI	IO	LINK интерфейс 0. Вход сигнала готовности
208	PC[26]	L0BCMPO	IO	LINK интерфейс 0. Подтверждение окончания обмена передатчиком
209	PC[27]	L0BCMPI	IO	LINK интерфейс 0. Подтверждение окончания обмена для приемника
210	PC[28]	L1ACKO	IO	LINK интерфейс 1. Выход сигнала готовности
211	PC[29]	L1ACKI	IO	LINK интерфейс 1. Вход сигнала готовности
212	PC[30]	L1BCMPO	IO	LINK интерфейс 1. Подтверждение окончания обмена передатчиком
213	VDD_CORE	–	PWR	Питание ядра
214	L1DATON[0]	–	O	LINK интерфейс 1. Выход данных (отрицательный), бит 0
215	L1DATOP[0]	–	O	LINK интерфейс 1. Выход данных (положительный), бит 0
216	L1DATON[1]	–	O	LINK интерфейс 1. Выход данных (отрицательный), бит 1

**Спецификация 1967ВЦ3Т, К1967ВЦ3Т, К1967ВЦ3ТК**

Номер вывода	Обозначение вывода	Дополнительное назначение вывода	Тип вывода	Функциональное назначение
217	L1DATOP[1]	–	O	LINK интерфейс 1. Выход данных (положительный), бит 1
218	VSS	–	GND	Общий
219	PC[31]	L1BCMPI	IO	LINK интерфейс 1. Подтверждение окончания обмена для приемника
220	L1DATON[2]	–	O	LINK интерфейс 1. Выход данных (отрицательный), бит 2
221	L1DATOP[2]	–	O	LINK интерфейс 1. Выход данных (положительный), бит 2
222	L1DATON[3]	–	O	LINK интерфейс 1. Выход данных (отрицательный), бит 3
223	L1DATOP[3]	–	O	LINK интерфейс 1. Выход данных (положительный), бит 3
224	VDD_IO	–	PWR	Питание контактных площадок
225	L1DATON[4]	–	O	LINK интерфейс 1. Выход данных (отрицательный), бит 4
226	L1DATOP[4]	–	O	LINK интерфейс 1. Выход данных (положительный), бит 4
227	L1DATON[5]	–	O	LINK интерфейс 1. Выход данных (отрицательный), бит 5
228	L1DATOP[5]	–	O	LINK интерфейс 1. Выход данных (положительный), бит 5
229	L1DATON[6]	–	O	LINK интерфейс 1. Выход данных (отрицательный), бит 6
230	L1DATOP[6]	–	O	LINK интерфейс 1. Выход данных (положительный), бит 6
231	L1DATON[7]	–	O	LINK интерфейс 1. Выход данных (отрицательный), бит 7
232	L1DATOP[7]	–	O	LINK интерфейс 1. Выход данных (положительный), бит 7
233	VDD_CORE	–	PWR	Питание ядра
234	L1CLKOUTN	–	O	LINK интерфейс 1. Выход синхросигнала (отрицательный)
235	L1CLKOUTP	–	O	LINK интерфейс 1. Выход синхросигнала (положительный)
236	L1DATIN[0]	–	I	LINK интерфейс 0. Вход данных (отрицательный), бит 0
237	L1DATIP[0]	–	I	LINK интерфейс 0. Вход данных (положительный), бит 0
238	VSS	–	GND	Общий

**Спецификация 1967ВЦ3Т, К1967ВЦ3Т, К1967ВЦ3ТК**

Номер вывода	Обозначение вывода	Дополнительное назначение вывода	Тип вывода	Функциональное назначение
239	L1DATIN[1]	–		LINK интерфейс 0. Вход данных (отрицательный), бит 1
240	L1DATIP[1]	–		LINK интерфейс 0. Вход данных (положительный), бит 1
241	L1DATIN[2]	–		LINK интерфейс 0. Вход данных (отрицательный), бит 2
242	L1DATIP[2]	–		LINK интерфейс 0. Вход данных (положительный), бит 2
243	L1DATIN[3]	–		LINK интерфейс 0. Вход данных (отрицательный), бит 3
244	L1DATIP[3]	–		LINK интерфейс 0. Вход данных (положительный), бит 3
245	L1DATIN[4]	–		LINK интерфейс 0. Вход данных (отрицательный), бит 4
246	L1DATIP[4]	–		LINK интерфейс 0. Вход данных (положительный), бит 4
247	L1DATIN[5]	–		LINK интерфейс 0. Вход данных (отрицательный), бит 5
248	L1DATIP[5]	–		LINK интерфейс 0. Вход данных (положительный), бит 5
249	L1DATIN[6]	–		LINK интерфейс 0. Вход данных (отрицательный), бит 6
250	L1DATIP[6]	–		LINK интерфейс 0. Вход данных (положительный), бит 6
251	L1DATIN[7]	–		LINK интерфейс 0. Вход данных (отрицательный), бит 7
252	L1DATIP[7]	–		LINK интерфейс 0. Вход данных (положительный), бит 7
253	L1CLKINN	–		LINK интерфейс 0. Вход синхросигналах (отрицательный)
254	L1CLKINP	–		LINK интерфейс 0. Вход синхросигнала (положительный)
255	nTRST	–		Интерфейс JTAG. Сброс (резистор доопределения до 1)
256	VSS	–	GND	Общий

## 4 Архитектура процессора ЦОС

Процессор состоит из трех архитектурных частей:

- ядро процессора (Рисунок 3), где исполняются команды;
- внутренняя память (Рисунок 4), где хранятся данные;
- периферийные устройства (Рисунок 5), которые осуществляют операции обмена с внешними устройствами.

В процессоре можно выделить следующие элементы:

- два вычислительных модуля: X и Y, каждый из которых содержит умножитель, ALU, CLU, сдвиговое устройство и регистровый файл объемом в 32 слова;
- два блока целочисленных ALU: J и K, каждый из которых содержит 32-битное целочисленное ALU, а также регистровый файл объемом в 32 слова;
- устройство управления (Sequencer), управляющее ходом исполнения программы и содержащее буфер выравнивания команд (instruction alignment buffer – IAB) и буфер целевых адресов перехода (branch target buffer – BTB);
- три 128-битные шины, обеспечивающие возможность высокоскоростного обмена между внутренней памятью и другими компонентами ядра процессора (вычислительными блоками, блоками целочисленных ALU, устройством управления и SOC-интерфейсом);
- 128-битная шина, обеспечивающая возможность высокоскоростного обмена между внутренней памятью и периферийными устройствами внешнего ввода/вывода (DMA, внешним портом и портами линков);
- SOC-интерфейс, обеспечивающий связь между внутренними шинами ядра и шиной периферийных устройств;
- периферийные устройства: интерфейс внешнего порта, контроллер SDRAM, конвейерный интерфейс со статической организацией конвейера, двенадцать каналов DMA, два порта LVDS-линков (с двумя каналами DMA каждый) и др.;
- 12 Мбит внутренней памяти, организованной как шесть блоков по 2 Мбит, каждый из которых содержит 64 К 32-битных слов;
- средства поддержки отладки;
- тестовый порт JTAG.

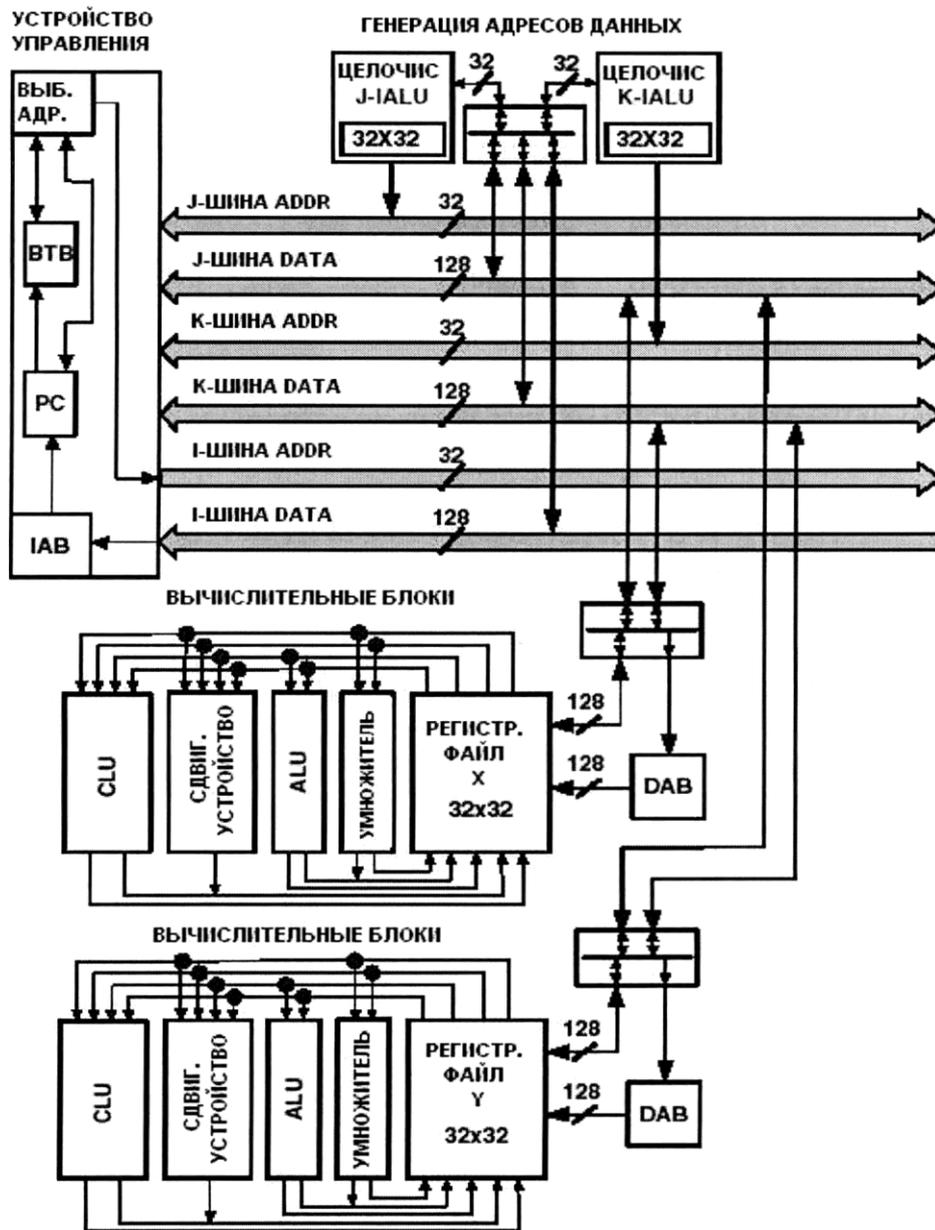


Рисунок 3 – Схема ядра процессора

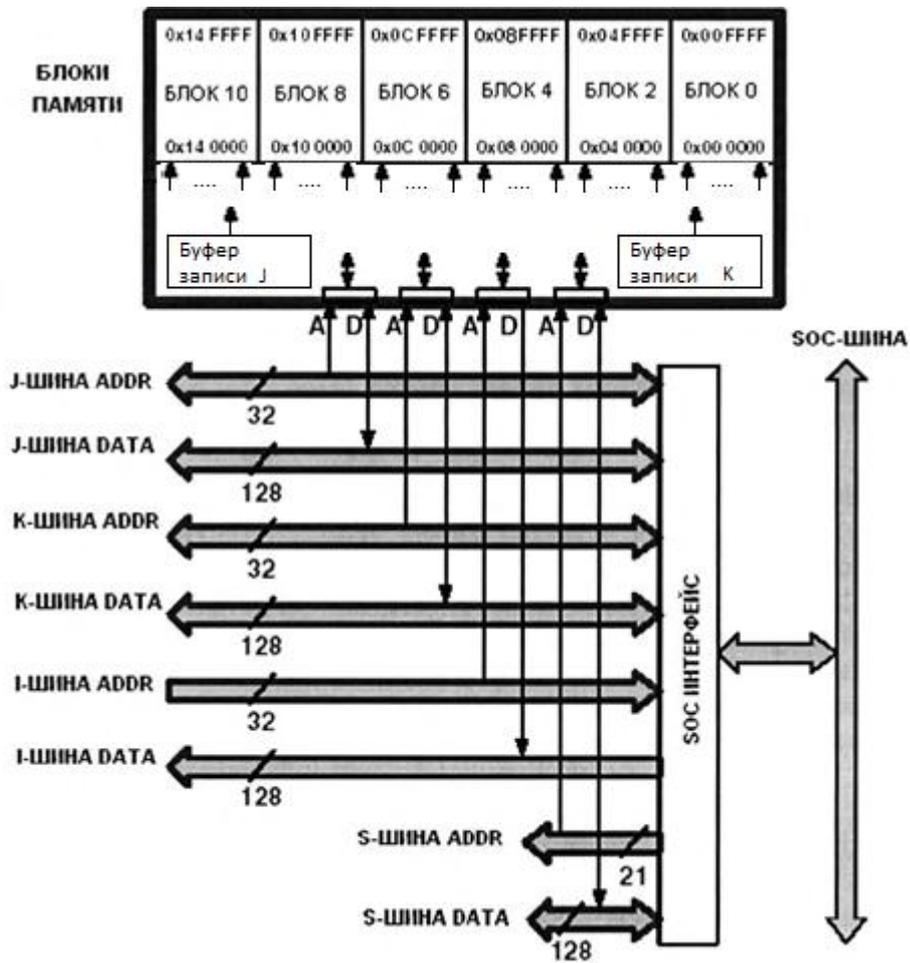


Рисунок 4 – Подключение внутренней памяти к ядру процессора

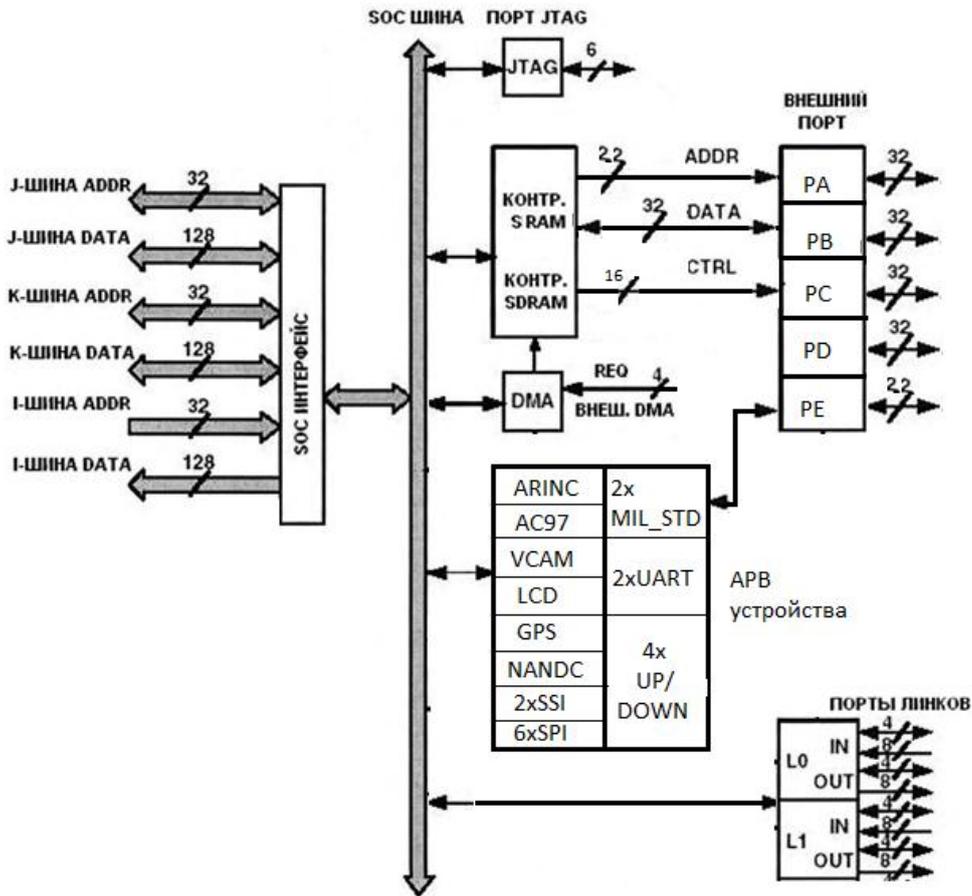


Рисунок 5 – Подключение периферийных устройств к ядру процессора

Внутренняя память объемом 12 Мбит разбита на шесть блоков памяти по 64 К слов. Каждая из четырех пар внутренних шин «адрес/данные» подсоединена ко всем шести блокам памяти через коммутационную матрицу. Шесть блоков памяти поддерживают до четырех обращений в каждом такте, причем каждый блок памяти может выполнить 128-битное обращение за такт.

Внешний порт поддерживает ширину шины данных 16 или 32 бита. Высокая пропускная способность ввода/вывода сочетается с высокой скоростью работы ядра. Для достижения высокой тактовой частоты процессор использует конвейерную внешнюю шину для синхронной статической памяти (SSRAM) и для синхронной динамической памяти (SDRAM).

Высокую пропускную способность передачи данных из точки в точку поддерживают два порта LVDS-линков. Каждый линк-порт обеспечивает полнодуплексную связь.

Наличие разнообразных контроллеров периферийных устройств позволяет подключать:

- внешнюю NAND флэш-память;
- последовательную флэш-память с интерфейсом SPI;
- внешние аудио кодеки с интерфейсом SSI или AC97;
- внешние устройства, поддерживающие интерфейсы ARINC и MIL\_STD;
- внешнюю LCD-панель произвольного разрешения;
- внешнюю видеокамеру;
- внешние ЦАП/АЦП.

## 4.1 Ядро процессора

В ядре процессора можно выделить пять самостоятельных функциональных модуля (Рисунок 3):

- Модули X и Y – вычислительные модули (устройства), в которых происходит основная обработка данных, и они в свою очередь включают в себя различные устройства обработки данных.
- Модули J и K – целочисленные АЛУ, которые также являются адресными генераторами (или адресными АЛУ), т.к. только они имеют возможность формирования адреса для доступа к памяти. Однако кроме адресных функций эти модули способны выполнять функции и по обработке данных. При этом данные (как и адреса) могут иметь только тип Integer, что соответствует 32-разрядному целому числу.
- Модуль S – устройство управления, которое формирует адреса команд, управляет потоком команд, а также управляет работой всего конвейера ядра.

Модули X и Y являются симметричными: все что делает модуль X, может делать и модуль Y. Каждый модуль обрабатывает свой поток команд, при этом модули X и Y можно рассматривать, как объединенный модуль, управляемый одной командой.

Модули J и K также симметричны, и каждый из них выполняет свой поток команд.

Устройство управления (модуль S) также имеет способность выполнять несколько потоков команд определенных типов. Среди них основными можно назвать команды переходов и вызовов.

Модульная архитектура процессора с независимыми потоками команд позволяет организовать высокопараллельные вычисления. Так в процессе интенсивных вычислений одно или оба целочисленных ALU вычисляют или генерируют адреса для выборки до двух операндов размером в квадрослово из двух блоков памяти, в то время как устройство управления одновременно извлекает следующую четверку команд из третьего блока памяти. Параллельно вычислительные устройства могут обрабатывать ранее считанные операнды, а устройство управления подготавливать переход.

Пока ядро процессора занято вышеописанными действиями, каналы DMA могут в фоновом режиме обновлять содержимое внутренней памяти квадрословами данных из внешнего порта или из портов линков.

Вычислительное ядро процессора достигает исключительно высокой производительности при цифровой обработке сигналов благодаря использованию следующих особенностей:

- вычислительного конвейера;
- пары вычислительных устройств;
- исполнения до четырех команд за такт;
- выборки/записи до восьми слов памяти за такт.

Два идентичных вычислительных устройства (X и Y) выполняют арифметические операции как с плавающей, так и с фиксированной точкой. Эти устройства выполняют до 6 операций с плавающей точкой или до 24 операций с фиксированной точкой за такт. Модули J и K выполняют SISD-обработку, модули X и Y могут выполнять SISD- и SIMD-обработку. При этом модули X и Y могут

управляться одной общей командой. Модули X и Y имеют сложную структуру и включают в себя разнообразные вычислительные блоки.

#### 4.1.1 Вычислительные модули

Ядро процессора содержит два вычислительных устройства, называемых *вычислительными модулями*. Каждый вычислительный модуль содержит регистровый файл и четыре независимых вычислительных блока: ALU, CLU, умножитель и сдвиговое устройство.

Вычислительные блоки способны обрабатывать данные в нескольких форматах представления с фиксированной и плавающей точкой (Рисунок 6):

- Форматы данных с фиксированной точкой:
- 64-битное длинное слово (Long Long);
- 32-битное обычное слово (Integer);
- 32-битное комплексное слово (16-битная действительная и 16-битная мнимая часть);
- 16-битное короткое слово (Short);
- 8-битное однобайтное слово (Char).

Для коротких слов арифметики с фиксированной точкой учетверенные операции над данными, выровненными на границу квадрослова, обеспечивают быструю обработку вектора данных. Байтовые операции поддерживаются также для данных, выровненных на границу октослова (т.е. 256-битного слова).

- Форматы данных с плавающей точкой:
- 32-битное обычное слово (float);
- 64-битное двойное слово (double);
- 40-битное расширенное слово.

Операции с плавающей точкой выполняются с одинарной, двойной и расширенной точностью. Формат обычного слова с плавающей точкой соответствует стандартному IEEE-формату, а 40-битное число формата с увеличенной точностью размещается в двойном слове (64 бита), занимая дополнительно к 32 битам восемь наименее значащих битов (Least Significant Bits – LSBs) мантиссы для достижения большей точности.

Каждый вычислительный модуль имеет многопортовый регистровый файл, содержащий регистры общего назначения для обмена данными между вычислительными блоками и шинами данных, а также для хранения промежуточных результатов. Ко всем этим регистрам можно обращаться как к обычному, двойному или квадрегистру. Все операции вычислительных модулей исполняются на двух стадиях конвейера. Результат операции доступен только на второй стадии, поэтому если следующая команда вычислительного модуля использует результат текущей команды как операнд-источник, всегда возникает такт простоя процессора (пузырь).

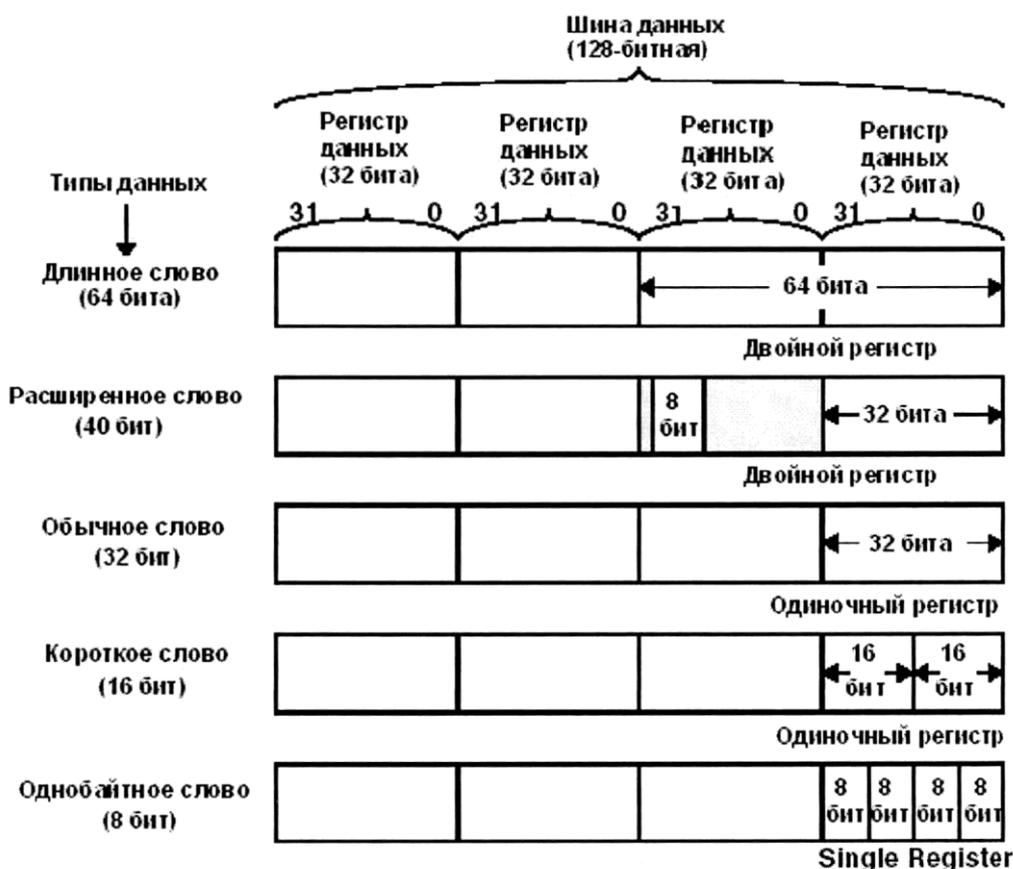


Рисунок 6 – Определения форматов слов

#### 4.1.1.1 Арифметико-логическое устройство (ALU)

ALU выполняет арифметические операции над данными с фиксированной и с плавающей точкой, а также логические операции над данными с фиксированной точкой. Источником и приемником большинства операций ALU является регистровый файл вычислительного блока. Операнды ALU могут иметь ширину 32 или 64 бита, что соответствует одиночным операциям над типами Int, Float, Double, Long Long. Для типов меньшей разрядности (Short, Char) поддерживается векторная обработка. Очевидно, что количество элементов в векторах для типа Int равно двум, для типа Short – двум или четырем, а для типа Char – четырем или восьми. Учитывая, что вычислительные модули X и Y могут работать как один удвоенный модуль, длина обрабатываемых векторов может быть увеличена в два раза, т.е. за такт процессор способен обрабатывать четыре типа Int, восемь типов Short или 16 типов Char. Все это соответствует максимальной ширине шины памяти в 128 бит.

#### 4.1.1.2 Коммуникационно-логическое устройство (CLU)

Каждый вычислительный модуль содержит вычислительное устройство специального назначения, называемое *коммуникационно-логическим устройством (CLU)*. Команды CLU предназначены для поддержки различных алгоритмов, используемых в коммуникационных приложениях, а именно алгоритмов:

- Декодирования по Витерби;
- Декодирования турбо кода;
- Сжатия [частотной полосы сигнала] (despreading) для систем множественного доступа с кодовым разделением (CDMA);
- Кросс-корреляций, используемых для отыскания пути.

Особенностью данного блока является собственный 32-словный набор регистров общего назначения. Таким образом, команды CLU могут использовать операнды из РОН вычислительного модуля и из РОН собственного блока.

#### **4.1.1.3 Умножитель с накоплением (Умножитель)**

Блок умножителя выполняет умножение чисел с фиксированной или плавающей точкой, а также операции умножения с накоплением для чисел с фиксированной точкой. Умножитель поддерживает несколько типов данных для форматов с фиксированной и плавающей точкой. Для плавающей точки – это стандартные форматы одинарной и двойной точности, а также формат с расширенной точностью. Источником и приемником для большинства операций является регистровый файл вычислительного блока. Умножитель также поддерживает операции умножения над комплексными числами, представленными в виде пары 16-разрядных коротких слов упакованными в одно 32-разрядное слово. Младшие значащие биты этого слова представляют собой действительную часть комплексного числа, а старшие значащие биты – мнимую часть. Команды умножения с накоплением используют в качестве приемника специальные регистры-аккумуляторы.

#### **4.1.1.4 Побитное сдвиговое устройство (сдвиговое устройство)**

Сдвиговое устройство выполняет логические и арифметические сдвиги, манипуляции с битами, заполнение полей и извлечение полей. Сдвиговое устройство выполняет операции с одним 64-битным, одним или двумя 32-битным, двумя или четырьмя 16-битными, а также с четырьмя или восемью 8-битными операндами с фиксированной точкой. Операции сдвигового устройства включают в себя:

- сдвиги и циклические сдвиги влево и вправо;
- операции манипуляции с битами, включая установку бита, очистку бита, переключение бита и проверку,
- операции манипуляции с битовыми полями, включая извлечение полей, заполнение полей с использованием регистра BFOTMP (внутреннего регистра сдвигового устройства);
- операции с одноканальным FIFO для поддержки потоков битов с полями переменной длины;
- поддержка операций преобразования фиксированной и плавающей точки (такими, как извлечение порядка, определение числа ведущих единиц и нулей).

#### **4.1.2 Целочисленное арифметико-логическое устройство (IALU)**

Целочисленные ALU могут исполнять стандартные автономные операции ALU с регистровыми файлами IALU, операции загрузки/сохранения регистров и пересылки из регистра в регистр, а также формировать адреса при обмене данными между памятью и регистрами. Процессор имеет пару IALU (J-IALU и K-IALU), что позволяет формировать адреса одновременно для двух параллельно исполняемых транзакций по 128 бит. Наличие целочисленных ALU позволяет вычислительным операциям исполняться с максимальной эффективностью, поскольку вычислительные устройства могут быть заняты исключительно обработкой данных.

Каждое IALU имеет многопортовый 32-словный регистровый файл. Любая вычислительная операция IALU выполняется за один такт. При генерации адресов целочисленные ALU поддерживают предмодификацию регистров без их обновления и постмодификацию с обновлением. Кольцевые буферы реализованы аппаратно. Целочисленные ALU поддерживают следующие типы команд:

- обычные команды IALU;
- команды пересылки данных [из регистра в регистр];
- команды загрузки константы в регистр;
- команды загрузки/сохранения данных с модификацией значением регистра;
- команды загрузки/сохранения данных с модификацией непосредственным значением.

При косвенной адресации (команды с модификацией) значение одного из регистров регистрового файла может быть модифицировано значением другого регистра файла или непосредственным 8- или 32-битным значением, как до (предмодификация), так и после (постмодификация) выполнения обращения к памяти. При адресации кольцевых буферов каждому из четырех первых регистров IALU может быть сопоставлено значение длины для выполнения автоматической адресации по ее модулю внутри такого буфера. Границами кольцевых буферов могут служить любые адреса памяти. Кольцевые буферы делают возможной эффективную реализацию линий задержки и других структур данных, используемых обычно в цифровых фильтрах и преобразованиях Фурье. Переход адресного указателя через границу кольцевого буфера обрабатывается процессором автоматически, что позволяет сократить накладные расходы и упростить программную реализацию алгоритмов.

Целочисленные ALU поддерживают также бит-реверсивную адресацию, ориентированную на алгоритм БПФ. Бит-реверсивная адресация реализуется с использованием сложения с обратным переносом, подобного обычному сложению, но у которого бит переноса берется из старших битов и направляется в младшие.

IALU обеспечивают гибкость в пересылках данных обычными, двойными или квадрословами. Каждая команда может исполняться с производительностью одна на такт. Обычно задержки, обусловленные зависимостью между командами IALU, отсутствуют, но если таковые имеются, их длительность может достигать трех тактов. Задержки могут появляться, если текущая команда производит загрузку данных из памяти в регистр IALU или пересылку регистра в регистр IALU, а следующая команда использует загружаемый регистр в качестве операнда-источника. В таком случае возникает три такта задержки до готовности операнда. Необходимо отметить, что IALU обладает практически такими же вычислительными возможностями по обработке данных типа Integer, как и вычислительный модуль. Наиболее проблемным местом является только отсутствие в IALU операции умножения.

#### **4.1.3 Устройство управления потоком команд**

Устройство управления (УУ) предоставляет адреса для выборки команд из памяти. УУ совместно с IALU позволяет вычислительным операциям выполняться с максимальной эффективностью. Эффективность ветвления обеспечивается устройством управления за счет использования буфера целевых адресов перехода (ВТВ), позволяющего сократить задержки на исполнение условных и безусловных переходов. УУ выполняет две задачи:

- декодирование извлеченных из памяти команд (выделение команд из полей командной строки) и отсылка их на соответствующие

исполнительные устройства (вычислительные блоки, целочисленные ALU или УУ);

- управление ходом исполнения программы.

Команды устройства управления подразделяются на два типа:

- *команды управления ходом исполнения программы* – используются для смены текущего адреса исполнения (перехода) и для условного исполнения индивидуальных команд;
- *команды непосредственного расширения* – используются для расширения числовых полей, указываемых в непосредственных операндах для УУ и IALU.

Команды управления ходом исполнения программы подразделяются, в свою очередь, на две категории:

- Явные переходы и вызовы, основанные на непосредственно указываемом в коде команды адресном операнде. Например, по команде 'if <cond> jump 100;' всегда осуществляется переход по адресу 100, если только вычисленное значение выражения <cond> есть «истина».
- Косвенные переходы, базирующиеся на адресе, предоставляемом регистром. (Команды, используемые для указания на условное исполнение командной строки, представляют собой подкатегорию косвенных переходов). Например, 'if <cond> cjmp;' является переходом по адресу, содержащемуся в регистре CJMP.

*Примечание* – Команды управления ходом исполнения должны располагаться в первом поле командной строки. Непосредственными расширениями могут обладать команды IALU и УУ (в последнем случае только команды управления ходом исполнения).

Команды управления ходом исполнения программы не задаются программистом явно, а порождаются автоматически в случае, если размер непосредственных данных, использованных в командах, достаточно велик. Программист должен поместить команду, которая требует непосредственного расширения, в первое поле командной строки и оставить свободное поле в этой строке (задействовав только три поля) с тем, чтобы ассемблер мог поместить непосредственное расширение во второе поле командной строки. В одной командной строке может присутствовать только одно непосредственное расширение.

Процессор достигает своей высокой скорости исполнения за счет использования 9-ти стадий конвейера. Каждая стадия соответствует одному такту частоты процессора. Упрощенная структура конвейера приведена ниже (Рисунок 7).

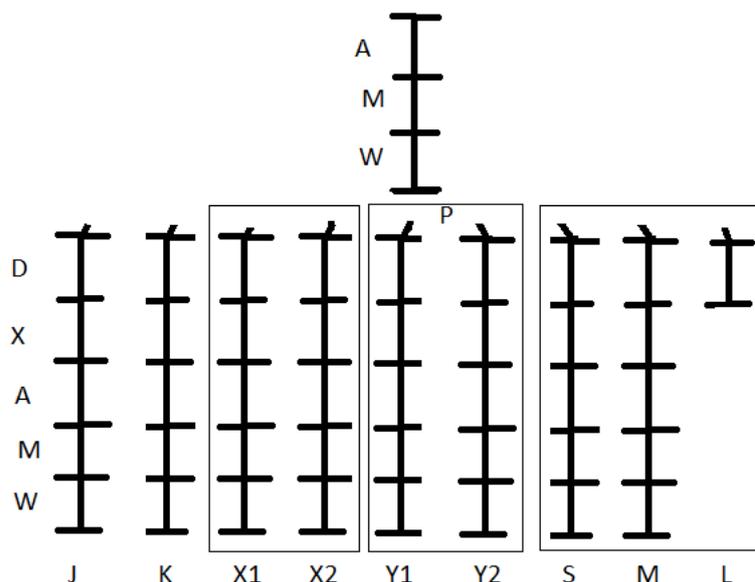


Рисунок 7 – Структура конвейера процессора

На рисунке горизонтальным линиям конвейера соответствуют регистры для хранения информации между стадиями конвейера, а вертикальные линии – задержке обрабатывающей логики. Самая верхняя горизонтальная линия соответствует регистру IP – указателю на текущую команду устройства управления. IP содержит адрес который отправляется в подсистему памяти для выборки команды.

Стадия адреса (стадия A) соответствует логике декодирования адреса, а также логике арбитража доступа к памяти.

Следующая стадия M соответствует чтению данных из накопителя в буфер памяти.

На стадии W происходит передача команды из буфера памяти в буфер выравнивания команд IAB. Шина команд имеет ширину 128 бит, что соответствует выборке 4-х команд за такт.

На стадии P одна линия конвейера *чтения* команд разветвляется на 9 линий (ветвей) конвейера *исполнения* команд.

Из рисунка видно, что модули J и K имеют по одной линии конвейера, модули X и Y – по две линии конвейера каждый, а модуль управления включает три ветви S, M, L. Ветвь L короткая, т.к. она обрабатывает команды-расширители константного поля, и вся обработка этой команды сводится к определению, какой линии конвейера соответствует расширение и передачи данных на эту линию. Для модулей обработки J, K, X, Y последняя горизонтальная линия конвейера соответствует РОН, в который записывается результат операции. Название стадий конвейера обработки данных D, X, A, M, W выбрано исходя из логики выполнения команды загрузки из памяти модулей J и K.

- D – декодирование команды;
- X – исполнение команды(вычисление адреса доступа);
- A – передача адреса в подсистему памяти;
- M – чтение данных из накопителя в буфер;
- W – передача данных из буфера памяти в регистр ядра.

В зависимости от типа команды операции, выполняемые на данных стадиях конвейера, будут различны. Так для модулей X и Y наиболее важными являются стадии A, M, W. На стадии A происходит декодирование команды, на стадии M – первый такт исполнения и на стадии W – второй (заключительный такт исполнения).

Одной из задач УУ является отслеживание зависимостей по операндам во всех линиях конвейера и приостановка конвейера в случае, если операнды не готовы. Все линии конвейера продвигаются синхронно. При этом максимум 6 линий из 9-ти могут исполнять одновременно команды.

Для УУ основной линией конвейера является линия S, на которой происходит обработка команд переходов. Для такого глубоко конвейеризированного процессора скорость выполнения переходов является одним из ключевых факторов эффективности исполнения программы. Переходы могут быть безусловными и условными. Условные переходы могут быть двух типов: переходы, анализируемые на стадии X конвейера и на стадии W. Последние (на W) соответствуют анализу флагов операций в модулях X и У. Можно видеть, что в случае выполнения перехода на стадии X процессор потеряет пять тактов, прежде чем следующая команда достигнет стадии X. Для ветвлений на стадии W таких тактов будет уже восемь. Для уменьшения потерь связанных с переходами, УУ содержит буфер целевых адресов перехода (ВТВ) и имеет механизм статического прогнозирования переходов (выполняется пользователем). Если переход помечен как прогнозируемый, он выполняется на стадии D линии S конвейера. Информация о переходе заносится в строку ВТВ. При первом выполнении данного перехода теряются четыре такта, однако при последующих его выполнениях потери будут равны 0.

Еще одной задачей УУ является обработка запросов прерываний.

Процессор имеет внешние линии запроса прерываний общего назначения, а также может обрабатывать прерывания от внутренних периферийных устройств. Есть возможность программным способом обеспечить обработку вложенных прерываний. Прерывания имеют низкую латентность и не могут сбросить с конвейера текущие команды, уже начавшие свое исполнение. Прерывание векторизуется непосредственно по заданному пользователем адресу в регистровом файле таблицы векторов прерываний.

Запрос прерывания поступает в УУ и если прерывания разрешены, стадия Р конвейера прекращает поставлять команды на линии обработки команд, а УУ помещает в регистр IP адрес вектора прерывания (сбрасывая содержимое конвейера чтения команд) и запускает чтение нового потока.

#### **4.1.4 Управление ядром процессора**

Существует несколько режимов работы процессора, а также средств управления его работой. В данном разделе описываются режимы работы, которые не управляются посредством выполнения команд. Эти средства управления включают управление:

- доменами синхронизации;
- режимом работы;
- режимом загрузки.

##### **4.1.4.1 Домены синхронизации**

Процессор использует для своей работы внешний вход тактовой частоты ХТ1. Данная частота используется в первую очередь в качестве опорной для внутреннего умножителя частоты (PLL). Также она используется внутренним RTC таймером. Структура формирования частот процессора показана ниже (Рисунок 8). Значение коэффициента умножения для PLL задается посредством программирования соответствующих регистров конфигурации. Модуль PLL формирует на своем выходе частоту ССЛК, которая используется для работы ядра. Шина периферийных

устройств использует для своей работы частоту SOCCLK, которая равна частоте ядра деленной на два. Дополнительно для работы каналов связи необходимо сформировать специальные частоты синхронизации, так как скорости их работы программируемые. Периферийные устройства используют в качестве базовой частоты частоту SOCCLK. Далее, внутри периферийного устройства, с помощью регистров конфигурации, может быть получена требуемая частота работы устройства.

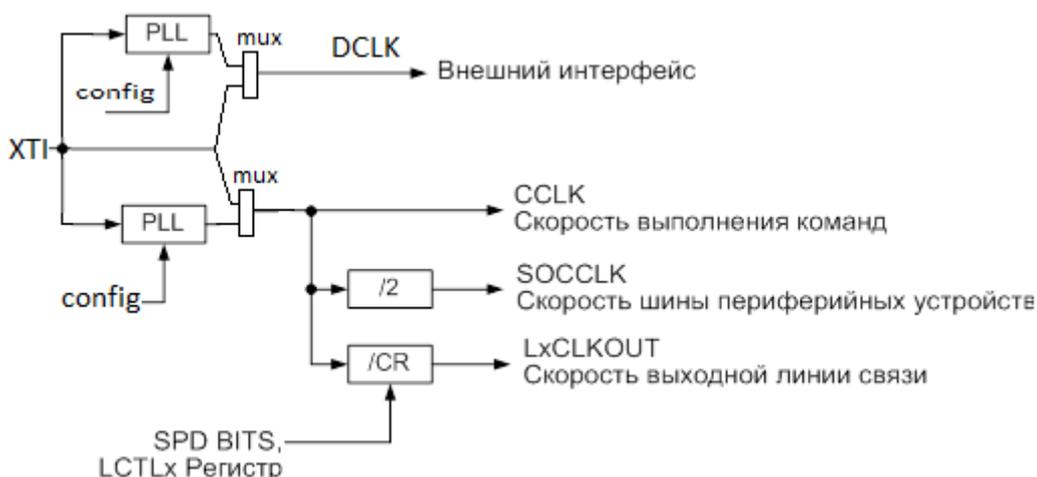


Рисунок 8 – Источники тактовой частоты

Для синхронизации внешнего интерфейса используется частота DCLK. После сброса модули PLL отключены, и процессор начинает свою работу на частоте XT1. Далее программа должна задать необходимые параметры частоты работы для PLL, выждать требуемое время необходимое для стабилизации выходной частоты и затем переключиться на частоту PLL.

Таким образом, в процессоре можно выделить три основных домена синхронизации:

- ядро процессора (CCLK);
- SOC-шина периферийных устройств (SOCCLK);
- внешний интерфейс (DCLK).

Дополнительными доменами синхронизации можно считать два порта связи, каждый из которых может иметь индивидуальную скорость передачи.

#### 4.1.4.2 Режимы работы

Процессор может работать в одном из трех режимов:

- пользовательский;
- супервизор;
- эмулятор.

В режиме пользователя и супервизора все команды выполняются стандартным образом (т.е. путем чтения из памяти). Отличие режима пользователя от режима супервизора заключается только в ограничении доступа к отдельным компонентам системы для пользователя. Режим также влияет на прием и обработку исключений. Приоритеты режимов от наименьшего приоритета к наивысшему: пользовательский, супервизор, эмулятор.

### **Пользовательский режим**

Работа в пользовательском режиме используется для алгоритмов и управляющего кода, который не требует манипуляций с системными ресурсами. Многие системные ресурсы недоступны в пользовательском режиме. Если процессор пытается получить доступы к таким ресурсам, возникает исключение. Ядро ОС работает в режиме супервизора, но пользовательский код ограничен пользовательским режимом.

Следующие регистры доступны ядру программы в пользовательском режиме:

- группы регистров от 0x00 до 0x09 – регистры вычислительных модулей;
- группы регистров от 0x0C до 0x0F – регистры целочисленных АЛУ;
- регистры УУ – CJMP, регистры счетчика циклов LC0, LC1 и регистр статичного флага (SFREG).

Все другие регистры не могут быть записаны ядром программы в пользовательском режиме. Попытка записи в защищенный регистр вызовет исключение.

### **Режим супервизора**

Режим супервизора позволяет программе получить доступ ко всем ресурсам процессора. Процессор работает в режиме супервизора при выполнении одного из условий:

- Бит NMOD в SQCTL установлен;
- Процедура прерывания запущена.

При аппаратном или программном сбросе процессор переходит в состояние ожидания прерывания. При получении прерывания он переходит в состояние исполнения программы в режиме супервизора.

Если бит NMOD в SQCTL сброшен, процессор входит в пользовательский режим после выхода из прерывания, если только прерывание не является вложенным.

### **Режим эмулятора**

Режим эмулятора используется при управлении процессором отладочным инструментом через порт JTAG. Процессор входит в режим эмулятора при генерации специального исключения. Исключение эмулятора генерируется при одном из событий:

- EMUTRAP команда;
- Срабатывание точки наблюдения для входа в эмулятор;
- Передний фронт TMS при установленном бите TEME в EMUCTL.

Исключения эмулятора имеют высочайший приоритет среди исключений и прерываний.

Пока процессор находится в режиме эмулятора, единственным источником команд для него является регистр EMUIR. Регистр EMUIR загружается через порт тестового доступа JTAG. При вхождении в режим эмулятора, внешний контроллер JTAG должен быть включен.

После работы в режиме эмулятора, единственным способом выхода из данного режима является выполнение команды возврата из прерывания (RTI).

В режиме эмулятора, доступ к регистрам отладки (группа регистров 0x1B) может быть получен путем выполнения команд пересылки «регистр-регистр» или командой непосредственной загрузки данных. Эти регистры не могут быть загружены из памяти.

ВТВ, счетчик циклов, мониторы производительности и буфер трассировки неактивны в режим эмулятора. Кроме команды RTI (np), все управляющие команды (переход, вызов, RDS, условные переходы, и т.д.) и команда IDLE не должны быть использованы в режиме эмулятора. Команда RTI может быть использована только безусловно.

#### **4.1.4.3 Режимы старта**

После сброса процессор начинает работу с выполнения специальной загрузочной программы, записанной во внутреннее ПЗУ. Данная программа осуществляет анализ внешних конфигурационных выводов и выполняет заданную последовательность действий. Подробное описание программы начального старта будет приведено в подразделе «Начальный старт процессора».

## 5 Память, регистры и шины

Глава содержит описание карты памяти и регистров. Для получения информации об использовании регистров для вычислений и памяти для загрузки и хранения необходимо обращаться к руководству по программированию процессора. Для информации по использованию регистров для конфигурирования процессорной периферии необходимо смотреть соответствующие главы описания периферийных модулей.

Встроенная память состоит из шести блоков по 2 Мб каждый. Каждый блок состоит из 64 К слов по 32 бита. Блоки памяти могут хранить команды и/или данные. Запросы к памяти обрабатываются конвейерно в пределах одного тактового цикла. Каждый запрос может извлекать до четырех слов.

Ядро процессора имеет три шины (I-шина, J-шина, и K-шина), каждая из которых соединена со всеми блоками внутренней памяти. Эти шины имеют ширину 128 бит для передачи до четырех команд или четырех слов данных в пределах одного цикла. Внешние по отношению к ядру устройства (DMA, хост-процессор, другие процессоры кластера) используют S-шину для доступа к памяти. В каждом цикле возможен только один доступ к каждому блоку памяти, т.е. если в течение одного такта мы имеем четыре запроса к различным модулям памяти, все запросы будут обслужены без потери скорости.

Большинство регистров процессора могут быть классифицированы как универсальные регистры (Ureg). Имеются команды для пересылки данных между двумя любыми Ureg, между Ureg и памятью или для непосредственной загрузки константы в Ureg.

Процессор имеет 32-битные адресные шины, обеспечивающие адресное пространство до четырех гигабайт. Процессор поддерживает словную адресацию памяти. *При этом также возможна и байтовая адресация.* Данная глава определяет карту памяти для процессора и показывает, где в пространстве памяти определяется каждый элемент. Зоны пространства памяти составлены из следующих областей:

- пространство внешней памяти – область для стандартной адресации внешней памяти, включая SDRAM, банк 0 (MS0), банк 1 (MS1)
- пространство внутренней памяти – область для стандартной внутренней адресации.

Общая карта памяти показана ниже (Рисунок 9).

Для быстродействия процессора большое значение имеет конфигурация банков внутренней памяти. Шесть банков внутренней памяти обеспечивают возможность одновременного параллельного доступа к ним в течение одного такта ядра. Такой уровень быстродействия можно достичь только, если процессор будет, например, производить чтение команд из банка 0, чтение данных по шине J из банка 1, чтение данных по шине K из банка 2, чтение-запись по шине S внешним мастером банка 3, отложенная запись по шине J в банк 4 и отложенная запись по шине K в банк 5.

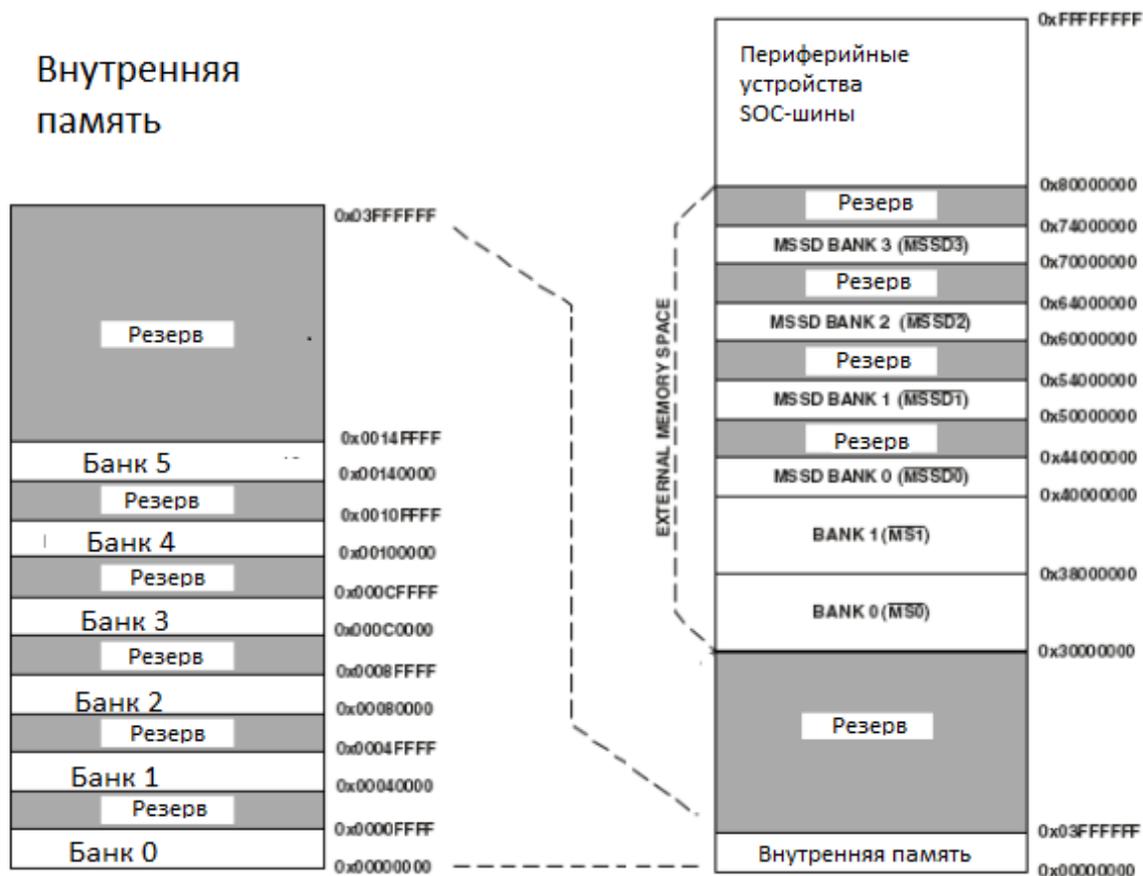


Рисунок 9 – Карта памяти

## 5.1 Пространство периферийных устройств SOC-шины

Пространство периферийных устройств SOC-шины используется процессором и контроллером DMA для адресации внутренних периферийных устройств, подключенных к SOC-шине. Если 31-й бит адреса равен 1, возможен доступ к данному адресному пространству. При этом доступ к регистрам некоторых периферийных устройств может поддерживаться непосредственно по имени (номеру) из команды процессора. Однако это возможно только к регистрам устройств, совместимых с процессором 1967ВЦ2Ф.

## 5.2 Пространство банка внешней памяти

Пространство банка внешней памяти относится к внешней памяти и устройствам ввода-вывода (SDRAM, SRAM, периферия ввода-вывода и другие стандартные устройства памяти). Внешний интерфейс поддерживает разбиение общего банка внешней памяти на отдельные банки и ставит им в соответствие специальные сигналы выборки и другие управляющие сигналы. Имеется набор банков для SDRAM, доступ к которым производится через контакты выбора внешней памяти MSSD0, MSSD1, MSSD2 и MSSD3. Доступ к этим банкам осуществляется по протоколу SDRAM. Другой набор банков определен контактами выбора внешней памяти MS0 и MS1, где протокол доступа конфигурируется пользователем как конвейерный или протокол медленного устройства, и параметры этих банков

определяются регистром SYSCON. Адрес внешней памяти разделяется на поля, как показано ниже (Таблица 2).

**Таблица 2 – Пространство банка внешней памяти**

<b>Биты</b>	<b>Имя</b>	<b>Определение</b>
ADDR25–0	Address	Биты, которые осуществляют выбор данных внутри адресного пространства определенного банка
ADDR30–26	MS/SD	Выбор типа банка внешней памяти: 01100 – банк 0 (MS0) 01110 – банк 1 (MS1) 10000 – SDRAM банк 0 (MSSD0) 10001 – зарезервировано 10100 – SDRAM банк 1 (MSSD1) 10101 – зарезервировано 11000 – SDRAM банк 2 (MSSD2) 11001 – зарезервировано 11100 – SDRAM банк 3 (MSSD3) 11101 – зарезервировано
ADDR31		Значение бита всегда будет равно нулю

### **5.3 Мультипроцессорное пространство**

В данном процессоре отсутствует поддержка многопроцессорности аналогичная процессору 1967ВЦ2Ф. Прежнее адресное пространство для мультипроцессорности зарезервировано.

### **5.4 Внутреннее адресное пространство**

Внутреннее адресное пространство (см. Рисунок 9) соответствует собственному адресному пространству процессора. Оно используется для передач внутри процессора, т.е. для доступа к блокам внутренней памяти.

### **5.5 Универсальные регистры**

К универсальным регистрам относятся все регистры вычислительного ядра. Универсальные регистры не отображаются в карте памяти, т.е. доступ к ним по адресу памяти невозможен. Доступ к регистрам возможен только при выполнении команд. Все регистры разбиты по группам. В одной группе находится 32 регистра. При явном доступе к регистру в коде команды указывается номер группы и номер регистра в группе. Регистры распределены по группам в соответствии с их отношением к различным модулям процессора. Например, регистры общего назначения (РОН) данных (XR31–0, YR31–0, или XYR31–0) вычислительных блоков находятся в одной группе регистров, тогда как РОН данных для целочисленного АЛУ (J30–0 или K30–0) – в других. Термин «универсальный регистр» указывает на следующие особенности.

Во-первых, содержимое регистра может быть загружено из памяти, сохранится в памяти или передаваться в/из других универсальных регистров процессора.

Во-вторых, регистр может быть доступен в одинарный, двойной и счетверенный (квадрорегистр). Ряд регистров может быть доступен только как одинарные.

Загрузка, сохранение и пересылка содержимого универсальных регистров доступна внутри процессора с использованием имени регистра в командах процессора, а не по адресу в памяти. Имена регистров содержатся в таблице групп регистров. Команда загрузки регистра читает данные из памяти процессора и помещает их в регистр. Команда сохранения регистра берет данные из регистра и помещает их в память процессора. Команда пересылки копирует данные из одного регистра в другой. Имеются также команды загрузки константы в регистр, когда значение операнда берется из кода команды и помещается в регистр. Следующий пример показывает доступы загрузки, сохранения и передачи одного регистра.

```
XR0 = 0x76543210 ;; /* загружает в XR0 конкретные 32-бит данные */  
XR4 = [J31 + 0x43] ;; /* загружает в XR0 данные из ячейки памяти с адресом 0x43 */  
[J0 + J4] = YR0 ;; /* сохраняет YR0 в память */  
XR7 = SQSTAT ;; /* передает содержимое SQSTAT в XR7 */
```

При возможности выполнения процессором до четырех команд, одновременно могут быть выполнены:

- две команды загрузки регистров из памяти
- две команды сохранения в память
- две команды пересылки.
- две загрузки регистров константами.

Все эти типы команд выполняются в целочисленных J и K модулях.

Для одновременного доступа к двум или четырем регистрам, образующим длинные слова и квадрослова, предусмотрена специальная техника образования нового имени регистра. При этом изменение имени регистра указывает на диапазон регистров, а не на новый регистр. Например, имя сдвоенного регистра XR1:0 указывает регистр R1 и R0 в вычислительном блоке X, а счетверенное имя регистра J31:28 указывает регистры J31, J30, J29, и J28 в J-IALU. Обязательно выравнивание границ адресов сдвоенных и счетверенных регистров. Для сдвоенного регистра его младший номер должен быть кратен двум (например, R1:0, R3:2, R27:26), а для счетверенного кратен 4 (например, R3:0, R7:4, R31:28). Следующий пример кода показывает загрузку, сохранение и передачу сдвоенного и счетверенного регистра.

```
YR5:4 = L [J31 + 0xf0] ;; /* загрузка YR5:4 из памяти */  
XR3:0 = Q [K4 += K5] ;; /* загрузка XR3:0 из памяти */  
L [J0 + J2] = YR31:30 ;; /* сохранение YR31:30 в память */  
Q [K31 + K28] = YR7:4 ;; /* сохранение YR7:4 в память */  
DCS0 = YR11:8 ;; /* передача YR11:8 в DCS0 (DP, DY, DX, DI) */
```

Обычно цикл загрузки или сохранения регистра в памяти происходит между регистром и словом памяти одинакового размера – 32, 64 и 128 бит. В этом случае операция называется «нормальный доступ чтения\записи».

Для использования преимуществ обработки «одна команда – много данных» (SIMD) архитектура шины процессора поддерживает типы доступа, в которых содержимое одного слова памяти может быть загружено в два регистра (широковещательное чтение), и тип доступа, в котором несколько регистров могут быть загружены\сохранены с различными данными, используя одно слово в памяти (доступ объединенного чтения или записи).

Пространство регистров состоит из 64 регистровых групп с количеством регистров в группе равным 32. Группы регистров определены в диапазоне 0x3F–0 (63–0). Группы 0x1F–0 (31–0) доступны для любых команд передачи данных (загрузка константой, пересылка регистра, загрузка и сохранение в памяти). Эти группы относятся к вычислительному ядру процессора.

Группы 0x3F–0x20 доступны только командам пересылки регистров и соответствуют регистрам некоторых периферийных устройств SOC-шины. Доступ к этим регистрам возможен как с помощью команд пересылки, так и с помощью адреса в пространстве адресов периферийных устройств.

Группы регистров вычислительного ядра:

- 0x00–0x09 вычислительные модули X и Y;
- 0x0C–0x0F целочисленные J и K АЛУ;
- 0x0A, 0x1B модуль отладки;
- 0x1A устройство управления;
- 0x1E– 0x1F устройство защиты памяти.

Группы регистров периферийных устройств.

- 0x20 – 0x23 контроллер DMA;
- 0x24 контроллер внешнего порта;
- 0x25 – 0x27 порты связи;
- 0x38, 0x39, и 0x3A контроллер прерываний;
- 0x3D модуль JTAG.

Прочие группы зарезервированы и не могут быть доступны для приложений, т.к. они могут вызвать непредсказуемую реакцию процессора.

Важно обратить внимание на номер регистра в плане его (регистра) расположения в процессоре: в ядре процессора (внутренние универсальные регистры) или в периферийном модуле SOC-шины, т.к. регистры имеют различные ограничения доступа. Некоторые из регистров не используют все 32 бита. Неиспользованные биты зарезервированы. При записи в регистр с резервными битами резервные биты должны быть записаны нулевым значением. При чтении регистра с резервными битами резервные биты могут иметь любое значение.

Ниже приведен перечень групп регистров вычислительного ядра. Описание групп регистров периферийных модулей приведено в главах, посвященных конкретным периферийным устройствам.

## **5.6 Группы регистров вычислительных модулей**

Файл регистров вычислительного модуля содержит 32 регистра общего назначения (РОН). Существует два подобных файла, по одному в каждом вычислительном модуле (X и Y). Однако, фактически имея две группы по 32 регистра, регистры вычислительных модулей используют 10 номеров групп (с 0 по 9) для своей адресации. Можно сказать, что каждый регистр имеет собственное имя и несколько псевдонимов. Использование подобных псевдонимов удобно в командах пересылки регистров, а также в командах чтения/записи при работе с памятью. С помощью псевдонима происходит дополнительное кодирование выполняемой с регистром операции.

Ниже указаны виды доступов к регистрам вычислительного блока в соответствии с их распределением в память (Таблица 3 – Таблица 7). Номер регистра образуется конкатенацией номера группы (старшие 6 бит) и номера регистра в группе (младшие 5 бит).

**Таблица 3 – Группа регистров вычислительного блока X**

<b>Имя</b>	<b>Номер</b>	<b>Значение по умолчанию</b>	<b>Имя</b>	<b>Номер</b>	<b>Значение по умолчанию</b>
XR0	0x 0000	Не определено	XR16	0x0010	Не определено
XR1	0x 0001	Не определено	XR17	0x 0011	Не определено
XR2	0x 0002	Не определено	XR18	0x 0012	Не определено
XR3	0x 0003	Не определено	XR19	0x 0013	Не определено
XR4	0x 0004	Не определено	XR20	0x 0014	Не определено
XR5	0x 0005	Не определено	XR21	0x 0015	Не определено
XR6	0x 0006	Не определено	XR22	0x 0016	Не определено
XR7	0x 0007	Не определено	XR23	0x 0017	Не определено
XR8	0x 0008	Не определено	XR24	0x 0018	Не определено
XR9	0x 0009	Не определено	XR25	0x 0019	Не определено
XR10	0x 000A	Не определено	XR26	0x 001A	Не определено
XR11	0x 000B	Не определено	XR27	0x 001B	Не определено
XR12	0x 000C	Не определено	XR28	0x 001C	Не определено
XR13	0x 000D	Не определено	XR29	0x 001D	Не определено
XR14	0x 000E	Не определено	XR30	0x 001E	Не определено
XR15	0x 000F	Не определено	XR31	0x 001F	Не определено

**Таблица 4 – Группа регистров вычислительного блока Y**

<b>Имя</b>	<b>Номер</b>	<b>Значение по умолчанию</b>	<b>Имя</b>	<b>Номер</b>	<b>Значение по умолчанию</b>
YR0	0x 0040	Не определено	YR16	0x 0050	Не определено
YR1	0x 0041	Не определено	YR17	0x 0051	Не определено
YR2	0x 0042	Не определено	YR18	0x 0052	Не определено
YR3	0x 0043	Не определено	YR19	0x 0053	Не определено
YR4	0x 0044	Не определено	YR20	0x 0054	Не определено
YR5	0x 0045	Не определено	YR21	0x 0055	Не определено
YR6	0x 0046	Не определено	YR22	0x 0056	Не определено
YR7	0x 0047	Не определено	YR23	0x 0057	Не определено
YR8	0x 0048	Не определено	YR24	0x 0058	Не определено
YR9	0x 0049	Не определено	YR25	0x 0059	Не определено
YR10	0x 004A	Не определено	YR26	0x 005A	Не определено
YR11	0x 004B	Не определено	YR27	0x 005B	Не определено
YR12	0x 004C	Не определено	YR28	0x 005C	Не определено
YR13	0x 004D	Не определено	YR29	0x 005D	Не определено
YR14	0x 004E	Не определено	YR30	0x 005E	Не определено
YR15	0x 004F	Не определено	YR31	0x 005F	Не определено

**Таблица 5 – Объединенная группа регистров вычислительного модуля ХУ**

<b>Имя</b>	<b>Адрес</b>	<b>Значение по умолчанию</b>	<b>Имя</b>	<b>Адрес</b>	<b>Значение по умолчанию</b>
XYR0	0x 0080	Не определено	XYR16	0x 0090	Не определено
XYR1	0x 0081	Не определено	XYR17	0x 0091	Не определено
XYR2	0x 0082	Не определено	XYR18	0x 0092	Не определено
XYR3	0x 0083	Не определено	XYR19	0x 0093	Не определено
XYR4	0x 0084	Не определено	XYR20	0x 0094	Не определено
XYR5	0x 0085	Не определено	XYR21	0x 0095	Не определено
XYR6	0x 0086	Не определено	XYR22	0x 0096	Не определено
XYR7	0x 0087	Не определено	XYR23	0x 0097	Не определено
XYR8	0x 0088	Не определено	XYR24	0x 0098	Не определено
XYR9	0x 0089	Не определено	XYR25	0x 0099	Не определено
XYR10	0x 008A	Не определено	XYR26	0x 009A	Не определено
XYR11	0x 008B	Не определено	XYR27	0x 009B	Не определено
XYR12	0x 008C	Не определено	XYR28	0x 009C	Не определено
XYR13	0x 008D	Не определено	XYR29	0x 009D	Не определено
XYR14	0x 008E	Не определено	XYR30	0x 009E	Не определено
XYR15	0x 008F	Не определено	XYR31	0x 009F	Не определено

**Таблица 6 – Объединенная группа регистров вычислительного модуля УХ**

<b>Имя</b>	<b>Адрес</b>	<b>Значение по умолчанию</b>	<b>Имя</b>	<b>Адрес</b>	<b>Значение по умолчанию</b>
YXR0	0x 00C0	Не определено	YXR16	0x 00D0	Не определено
YXR1	0x 00C1	Не определено	YXR17	0x 00D1	Не определено
YXR2	0x 00C2	Не определено	YXR18	0x 00D2	Не определено
YXR3	0x 00C3	Не определено	YXR19	0x 00D3	Не определено
YXR4	0x 00C4	Не определено	YXR20	0x 00D4	Не определено
YXR5	0x 00C5	Не определено	YXR21	0x 00D5	Не определено
YXR6	0x 00C6	Не определено	YXR22	0x 00D6	Не определено
YXR7	0x 00C7	Не определено	YXR23	0x 00D7	Не определено
YXR8	0x 00C8	Не определено	YXR24	0x 00D8	Не определено
YXR9	0x 00C9	Не определено	YXR25	0x 00D9	Не определено
YXR10	0x 00CA	Не определено	YXR26	0x 00DA	Не определено
YXR11	0x 00CB	Не определено	YXR27	0x 00DB	Не определено
YXR12	0x 00CC	Не определено	YXR28	0x 00DC	Не определено
YXR13	0x 00CD	Не определено	YXR29	0x 00DD	Не определено
YXR14	0x 00CE	Не определено	YXR30	0x 00DE	Не определено
YXR15	0x 00CF	Не определено	YXR31	0x 00DF	Не определено

**Таблица 7 – Группа регистров широковещательного доступа вычислительного модуля XY**

<b>Имя</b>	<b>Адрес</b>	<b>Значение по умолчанию</b>	<b>Имя</b>	<b>Адрес</b>	<b>Значение по умолчанию</b>
XYR0	0x 0100	Не определено	XYR16	0x 0110	Не определено
XYR1	0x 0101	Не определено	XYR17	0x 0111	Не определено
XYR2	0x 0102	Не определено	XYR18	0x 0112	Не определено
XYR3	0x 0103	Не определено	XYR19	0x 0113	Не определено
XYR4	0x 0104	Не определено	XYR20	0x 0114	Не определено
XYR5	0x 0105	Не определено	XYR21	0x 0115	Не определено
XYR6	0x 0106	Не определено	XYR22	0x 0116	Не определено
XYR7	0x 0107	Не определено	XYR23	0x 0117	Не определено
XYR8	0x 0108	Не определено	XYR24	0x 0118	Не определено
XYR9	0x 0109	Не определено	XYR25	0x 0119	Не определено
XYR10	0x 010A	Не определено	XYR26	0x 011A	Не определено
XYR11	0x 010B	Не определено	XYR27	0x 011B	Не определено
XYR12	0x 010C	Не определено	XYR28	0x 011C	Не определено
XYR13	0x 010D	Не определено	XYR29	0x 011D	Не определено
XYR14	0x 010E	Не определено	XYR30	0x 011E	Не определено
XYR15	0x 010F	Не определено	XYR31	0x 011F	Не определено

Отметим, что дополнительно имеются номера групп которые используются командами альтернативного доступа с использованием буфера DAB и кольцевых буферов.

## **5.7 Регистры вычислительного модуля без номера**

Есть несколько регистров в вычислительных модулях, которые не являются универсальными, поэтому не доступны посредством их адресации с помощью номера группы и номера в группе. Эти регистры доступны только при выполнении специальных команд, которые передают данные между ними и РОН вычислительного модуля.

### **5.7.1 Регистры статуса вычислительных модулей (XSTAT/YSTAT)**

Регистры XSTAT и YSTAT – 32-битные регистры вычислительных модулей, которые хранят состояние флагов модулей. Каждый флаг регистра обновляется при выполнении соответствующей команды. Вычислительный модуль состоит из вычислительных блоков: ALU, умножитель, сдвигатель, CLU. Каждый из этих блоков формирует специальные флаги-признаки результата операции. Регистр состояний хранит данные флаги. Флаги имеют разный тип. Большинство флагов регистра изменяет свое значение при выполнении соответствующих команд, однако есть специальные флаги (sticky), которые после их установки в 1-е значение уже невозможно сбросить. Сброс возможен только командой загрузки регистра. Sticky биты очень удобны, когда нужно в конце алгоритма проверить были ли аномальные ситуации при выполнении вычислений на протяжении всего алгоритма. В отличие от них другие флаги нужно проверять (если необходимо) сразу после выполнения

операции. Подробное описание бит регистров X/YSTAT (значение после сброса = 0x0000 0000) приведено ниже (Таблица 8).

**Таблица 8 – Регистр состояния вычислительного модуля**

<b>Бит</b>	<b>Имя</b>	<b>Назначение</b>
0	AZ	ALU. Признак нулевого результата операции
1	AN	ALU. Признак отрицательного результата операции
2	AV	ALU. Признак переполнения результата операции
3	AC	ALU. Признак переноса
4	MZ	Умножитель. Признак нулевого результата операции
5	MN	Умножитель. Признак отрицательного результата операции
6	MV	Умножитель. Признак переполнения результата операции
7	MU	Умножитель. Признак исчезновения (underflow) результата операции
8	SZ	Сдвигатель. Признак нулевого результата операции
9	SN	Сдвигатель. Признак отрицательного результата операции
10	BF0	Сдвигатель. Флаг блока операций с плавающей запятой
11	BF1	Сдвигатель. Флаг блока операций с плавающей запятой
12	AI	Ошибочная (Invalid) операция АЛУ с плавающей запятой
13	MI	Ошибочная (Invalid) операция умножителя с плавающей запятой
14	TROV	CLU. Переполнение операции trellis
15	TRSOV	CLU. Переполнение операции trellis. Sticky бит
16	-	
17	-	
18	-	
19	-	
20	UEN	Разрешение (если 1) генерации исключительной ситуации при возникновении флага исчезновения (underflow) результата
21	OEN	Разрешение (если 1) генерации исключительной ситуации при возникновении флага переполнения(overflow) результата
22	IVEN	Разрешение (если 1) генерации исключительной ситуации при возникновении флага ошибочной (invalid) результата при операциях с плавающей запятой
23	-	
<b>Sticky биты</b>		
24	AUS	Underflow в АЛУ с плавающей запятой
25	AVS	Переполнение в АЛУ с плавающей запятой
26	AOS	Переполнение в АЛУ с фиксированной точкой
27	AIS	Ошибочный результат в АЛУ с плавающей запятой
28	MUS	Underflow в умножителе с плавающей запятой
29	MVS	Переполнение в умножителе с плавающей запятой
30	MOS	Переполнение в умножителе с фиксированной точкой
31	MIS	Ошибочный результат в умножителе с плавающей запятой

### **5.7.2 Регистры АЛУ**

Регистры параллельных результатов (PR0 и PR1) – 32-битные регистры, используемые для специальных команд суммирования.

### **5.7.3 Регистры умножителя**

Регистры результатов умножителя (MR3–0 и MR4) используются для накопления при выполнении операций умножения с накоплением, т.е. выполняют функции аккумуляторов.

### 5.7.4 Регистры сдвигателя

Регистр VFOTMP – 64-битный регистр для команды PUTBITS, используемой для организации потока бит.

### 5.7.5 Регистры блока CLU

Модуль CLU использует дополнительный набор регистров для выполнения своих команд. Это регистры данных (TR31–0), регистры истории (THR3–0) и регистр управления (SMCTL). Каждый регистр имеют ширину 32 бита.

## 5.8 Группы регистров целочисленного АЛУ

Каждое целочисленное АЛУ имеет две группы универсальных регистров:

- 1-ая группа – файл регистров общего назначения, включающий 32 слова;
- 2-ая группа – регистры для указания параметров кольцевой буферизации.

Группы регистров целочисленного АЛУ показаны ниже (Таблица 9 – Таблица 12).

Файл регистров кольцевого буфера содержит следующие регистры:

- 3 – 0 – кольцевого буфера JB3–0 и KB3–0;
- 7 – 4 – длина кольцевого буфера JL3–0 и KL3–0;
- 26 – 8 – резерв;
- 27 – регистр указателя стека режима пользователя (если включено разрешение на его использование);
- 31 – 28 – резерв.

**Таблица 9 – Группа регистров целочисленного АЛУ типа J (J-IALU)**

Имя	Номер	Значение по умолчанию	Имя	Номер	Значение по умолчанию
J0	0x 0180	Не определено	J16	0x 0190	Не определено
J1	0x 0181	Не определено	J17	0x 0191	Не определено
J2	0x 0182	Не определено	J18	0x 0192	Не определено
J3	0x 0183	Не определено	J19	0x 0193	Не определено
J4	0x 0184	Не определено	J20	0x 0194	Не определено
J5	0x 0185	Не определено	J21	0x 0195	Не определено
J6	0x 0186	Не определено	J22	0x 0196	Не определено
J7	0x 0187	Не определено	J23	0x 0197	Не определено
J8	0x 0188	Не определено	J24	0x 0198	Не определено
J9	0x 0189	Не определено	J25	0x 0199	Не определено
J10	0x 018A	Не определено	J26	0x 019A	Не определено
J11	0x 018B	Не определено	J27	0x 019B	Не определено
J12	0x 018C	Не определено	J28	0x 019C	Не определено
J13	0x 018D	Не определено	J29	0x 019D	Не определено
J14	0x 018E	Не определено	J30	0x 019E	Не определено
J15	0x 018F	Не определено	J31	0x 019F	Не определено

**Таблица 10 – Группа регистров целочисленного АЛУ типа К (K-IALU)**

Имя	Номер	Значение по умолчанию	Имя	Номер	Значение по умолчанию
K0	0x 01A0	Не определено	K16	0x 01B0	Не определено
K1	0x 01A1	Не определено	K17	0x 01B1	Не определено
K2	0x 01A2	Не определено	K18	0x 01B2	Не определено
K3	0x 01A3	Не определено	K19	0x 01B3	Не определено
K4	0x 01A4	Не определено	K20	0x 01B4	Не определено
K5	0x 01A5	Не определено	K21	0x 01B5	Не определено
K6	0x 01A6	Не определено	K22	0x 01B6	Не определено
K7	0x 01A7	Не определено	K23	0x 01B7	Не определено
K8	0x 01A8	Не определено	K24	0x 01B8	Не определено
K9	0x 01A9	Не определено	K25	0x 01B9	Не определено
K10	0x 01AA	Не определено	K26	0x 01BA	Не определено
K11	0x 01AB	Не определено	K27	0x 01BB	Не определено
K12	0x01AC	Не определено	K28	0x 01BC	Не определено
K13	0x01AD	Не определено	K29	0x 01BD	Не определено
K14	0x 01AE	Не определено	K30	0x 01BE	Не определено
K15	0x 01AF	Не определено	K31	0x 01BF	Не определено

**Таблица 11 – Группа регистров циклического буфера целочисленного АЛУ типа J (J-IALU)**

Имя	Номер	Значение по умолчанию
JB0	0x 01C0	Не определено
JB1	0x 01C1	Не определено
JB2	0x 01C2	Не определено
JB3	0x 01C3	Не определено
JL0	0x 01C4	Не определено
JL1	0x 01C5	Не определено
JL2	0x 01C6	Не определено
JL3	0x 01C7	Не определено
резервные	0x 01C8– 0x 01DA	Не определено
<b>JUSP</b>	0x 01DB	Не определено
резервные	0x 01DC– 0x 01DF	Не определено

**Таблица 12 – Группа регистров циклического буфера K-IALU**

Имя	Номер	Значение по умолчанию
KB0	0x 01E0	Не определено
KB1	0x 01E1	Не определено
KB2	0x 01E2	Не определено
KB3	0x 01E3	Не определено
KL0	0x 01E4	Не определено
KL1	0x 01E5	Не определено
KL2	0x 01E6	Не определено
KL3	0x 01E7	Не определено
резервные	0x 01E8– 0x 01FA	Не определено
KUSP	0x 01FB	Не определено
резервные	0x 01FC– 0x 01FF	Не определено

### 5.8.1 Регистры статусы целочисленного АЛУ (J31/JSTAT и K31/KSTAT)

Регистры статуса JSTAT (для J-IALU) и KSTAT (для K-IALU) хранят состояния флагов и обновляются в результате выполнения различных команд целочисленного АЛУ. При записи возможно обращение к регистру JSTAT как J31. При использовании в качестве операнда в арифметических, логических и функциональных операциях J31 рассматривается как ноль. Регистры J31/JSTAT (K31/KSTAT) (значение сброса = 0x0000 0000) описаны ниже (Таблица 13).

**Таблица 13 – Регистр состояния целочисленного АЛУ**

Бит	Имя	Назначение
0	Z	Признак нулевого результата операции
1	N	Признак отрицательного результата операции
2	V	Признак переполнения результата операции
3	C	Признак переноса
4	-	Резервные. Всегда 0

## 5.9 Группы регистров устройства управления

Группа содержит 32 регистра, однако используются не все регистры. Часть регистров зарезервирована. Все регистры группы выполняют специальные функции. Особенностью данной группы является то, что регистры доступны только как однословные.

**Таблица 14 – Группа регистров устройства управления**

Имя	Описание	Номер	Значение по умолчанию
CJMP	Вычисленный адрес перехода	0x 0340	Не определено
-	-	0x 0341	-
RETI	Возврат из прерывания	0x 0342	Не определено
RETIB	Псевдоним RETI, для вложенных прерываний	0x 0343	Не определено
RETS	Возврат из исключительной ситуации	0x 0344	Не определено
DBGE	Возврат из эмуляции	0x 0345	Не определено
-	-	0x 0346– 0x 0347	-
LC0	Счетчик циклов 0	0x 0348	Не определено
LC1	Счетчик циклов 1	0x 0349	Не определено
-	-	0x 034A– 0x 034F	-
IVSW	Адрес-вектор для программных исключений	0x 0350	Не определено
-	-	0x 0351– 0x 0353	-
FLAGREG	Регистр управления флагами процессора	0x 0354	0x0000 0000
FLAGREGST	Установка бит регистра флага	0x 0355	Не определено
FLAGREGCL	Сброс бит регистра флага	0x 0356	Не определено
-	-	0x 0357	-
SQCTL	Регистр управления	0x 0358	0x0000 0004
SQCTLST	Установка бит регистра управления	0x 0359	Не определено
SQCTLCL	Сброс бит регистра управления	0x 035A	Не определено
SQSTAT	Регистр статуса, только чтение	0x 035B	0x0000 FF04

SFREG	регистр специальных статических флагов	0x 035C	0x0000 0000
-	-	0x 035D	-
-	-	0x 035E	-
-	-	0x 035F	-

### 5.9.1 Регистр управления флагом (FLAGREG)

Регистр FLAGREG – 32-битный регистр, который контролирует направление контакта флага (вход или выход) и обеспечивает значение на контакте (1 или 0), когда контакт сконфигурирован как выход. Регистр FLAGREG (значение сброса = 0x0000 0000) описывается ниже (Таблица 15).

**Таблица 15 – Биты регистра управления флагом**

Бит	Имя	Назначение
3:0	FLAGx_EN	Направление флага 0 – прием 1 – выдача
7:4	FLAGx_OUT	Значение флага при выдаче
31-8	-	Не используются и всегда равны нулю

Контакт FLAG0 использует биты 0 и 4, FLAG1 использует биты 1 и 5, FLAG2 использует биты 2 и 6, FLAG3 использует биты 3 и 7.

### 5.9.2 Регистр управления SQCTL

Устройство управления последовательностью команд контролируется значением регистра SQCTL. После сброса значение регистра равно 0x0000\_0204. Подробное описание всех бит регистра приведено ниже (Таблица 16).

**Таблица 16 – Биты регистра управления SQCTL**

Бит	Имя	Назначение
0	-	
1	-	
2	GIE	Глобальное разрешение прерываний: 1 – прерывания разрешены; 0 – запрещены
3	SWIE	Разрешение генерации исключительных ситуаций (программных прерываний): 1 – разрешено; 0 – запрещено
4	-	
5	-	
6	-	
7	-	
8	DBGEN	Разрешение отладки: 1 – разрешено; 0 – запрещено
9	NMOD	Режим работы: 0 – пользователь; 1 – супервизор
10	TRCBEN	Включение буфера трасс: 0 – буфер трасс выключен и хранит старое значение; 1 – включен и следит за исполнением переходов

11	TRCBEXEN	Разрешение генерации исключительной ситуации от буфера трасс: 0 – запрещено; 1 – разрешено
12	DF_IEEE	Разрешение использования команд обработки данных с плавающей запятой двойной точности: 0 – используется расширенная точность; 1 – код команды расширенной точности соответствует коду команды двойной точности
13	BYTE_ON	Разрешение использования кодов команды TRAP 32-63 в качестве кодов префикса для создания новых команд: 0 – команды TRAP 32-63 используются для генерации исключений; 1 – команды TRAP 32-63 используются в качестве префикса.
14	USP	Разрешение использования двух указателей стеков (регистр JK27): 0 – регистр JK27 один для пользователя и супервизора; 1 – для пользователя и супервизора используются разные регистры
31-15	-	Зарезервированы и не используются. При чтении всегда равны нулю

Биты 12, 13, 14 регистра управления соответствуют дополнительным возможностям процессора, и их включение должно быть дополнительно разрешено.

### 5.9.3 Регистр управления (SQCTLST). Установка бит

Бит SQCTLST является псевдонимом для SQCTL. При записи по данному адресу '1' в любом бите записываемых данных устанавливает соответствующий бит в SQCTL, тогда как '0' в записываемых данных не меняет значение бита.

### 5.9.4 Регистр управления (SQCTLCL). Сброс бит

Бит SQCTLCL является псевдонимом для записи в SQCTL. При записи по данному адресу '0' в любом бите записываемых данных сбрасывает соответствующий бит в SQCTL, тогда как '1' в записываемых данных не меняет значение бита.

### 5.9.5 Регистр статических флагов (SFREG)

Статические флаги используются как статические копии условий. При желании сохранить значение условия, до того как другая команда поменяет его значение, можно скопировать значение в регистр SFREG и использовать его позже как условие. Все статические флаги условий группируются в регистре SFREG. Регистр SFREG после сброса равен нулю и описание его разрядов приведено ниже (Таблица 17).

**Таблица 17 – Биты регистра статических флагов**

Бит	Имя	Назначение
0	GSCF0	Статический флаг целочисленных АЛУ
1	GSCF1	Статический флаг целочисленных АЛУ
2	XSCF0	Статический флаг вычислительного модуля X
3	XSCF1	Статический флаг вычислительного модуля X
4	YSCF0	Статический флаг вычислительного модуля У
5	YSCF1	Статический флаг вычислительного модуля У
31-6	-	Не используются и всегда равны нулю

### 5.9.6 Регистр статуса устройства управления (SQSTAT)

Данный регистр доступен только для чтения и содержит информацию о текущем статусе устройства управления. Подробное описание разрядов регистра приведено ниже (Таблица 18).

Таблица 18 – Биты регистра SQSTAT

Бит	Имя	Назначение
1:0	MODE	Текущий режим работы процессора: 00 – пользователь 01 – супервизор 11 – эмулятор Другие состояния невозможны
2	IDLE	Процессор в состоянии ожидания прерывания (если 1)
7:3	SPVCMD	Значение параметра последней выполненной команды TRAP
11:8	EXCAUSE	Код последней исключительной ситуации: 0000 – команда TRAP 0001 – точка наблюдения или буфер трасс 0010 – команда плавающей запятой 0011 – ошибочная линия команд 0100 – невыровненный доступ 0101 – попытка записи защищенного регистра 0110 – переполнения счетчика статистики 0111 – ошибочный доступ в IALU 1000 – x 1001 – x 1010 – x 1011 – x 1100 – x 1101 – x 1110 – x 1111 – не было ошибок с момента сброса
15:12	EMCAUSE	Код ситуации вызвавшей переход в режим эмулятора: 1111 – не было перехода в режим эмулятора после сброса 0000 – команда EMUTRAP 0001 – запрос JTAG 0010 – сработала точка наблюдения Все другие коды зарезервированы и не могут присутствовать
16	FLAG0	Состояние внешнего контакта FLAG0
17	FLAG1	Состояние внешнего контакта FLAG1
18	FLAG2	Состояние внешнего контакта FLAG2
19	FLAG3	Состояние внешнего контакта FLAG3
20	EXE_ISR	Флаг устанавливается в 1 при: - переходе процессора в состояние обработки прерывания; - записи данных в регистр RETIB.  Установленный флаг запрещает прерывания. Флаг сбрасывается: - командой выхода из прерывания RTI; - командой RDS (если нет обработки исключительной ситуации); - чтением регистра RETIB.  Сброшенный флаг означает возможность обработки нового прерывания процессором
21	EXE_SWI	Признак того что процессор находится в состоянии обработки

		исключительной ситуации (когда равен 1).
22	EMUL	Признак того что процессор находится в режиме эмулятора (когда равен 1)
23	-	
24	BTBEN	
25	-	BTBLK
26	-	
27	-	
28	BTBEN	Буфер предсказания переходов включен (если 1) или выключен (если 0).
29	-	BTBLK
30	-	
31	I_LOCK	Флаг блокировки прерываний (1 – прерывания заблокированы)

### 5.10 Группа регистров устройства защиты памяти

Выше упоминалось, что процессор имеет несколько режимов работы и особенностью режима пользователя является то, что для него ограничен доступ к некоторым аппаратным ресурсам. Однако внутренняя и внешняя память одинаково доступны как для супервизора, так и для пользователя. Вместе с тем в ряде приложений желательно защитить код или данные системы от разрушения некорректным поведением пользовательской программы. Для этих целей в процессоре имеется группа регистров, с помощью которой можно описать отдельные регионы внутренней памяти и способы доступа к ним. Также эта группа регистров содержит регистры управления кэш-памятью. Все регистры данной группы доступны только как однословные.

**Таблица 19 – Группа регистра устройства защиты памяти**

Имя	Описание	Номер	Значение по умолчанию
<b>Группа 0x1E</b>			
MS0_C	Регистр управления кэшированием данных MS0	0x03c0	0
MS0_WT	Регистр управления сквозной записью MS0	0x03c1	0
MS0_CI	Регистр управления кэшированием команд MS0	0x03c2	0
		0x03c3-0x03c7	
MS1_C	Регистр управления кэшированием данных MS1	0x03c8	0
MS1_WT	Регистр управления сквозной записью MS1	0x03c9	0
MS1_CI	Регистр управления кэшированием команд MS1	0x03ca	0
		0x03cb-0x03cF	
SDR_C	Регистр управления кэшированием данных SDRAM	0x03D0	0
SDR_WT	Регистр управления сквозной записью SDRAM	0x03D1	0
SDR_CI	Регистр управления кэшированием команд SDRAM	0x03D2	0
		0x03D3-0x03DF	
<b>Группа 0x1F</b>			
PU0	Регистр защиты 0	0x 03E0	Не определено

PU1	Регистр защиты 1	0x 03E1	Не определено
PU2	Регистр защиты 2	0x 03E2	Не определено
PU3	Регистр защиты 3	0x 03E3	Не определено
PU4	Регистр защиты 4	0x 03E4	Не определено
PU5	Регистр защиты 5	0x 03E5	Не определено
PU6	Регистр защиты 6	0x 03E6	Не определено
PU7	Регистр защиты 7	0x 03E7	Не определено
STS_PROT	Регистр состояния	0x 03E8	0
-	-	0x 03E9– 0x 03FB	0
PU_CR	Регистр управления	0x 03FC	0
IDC_STB	Управление кэш-памятью	0x 03FD	0
-	-	0x 03FE– 0x 03FF	0

### 5.10.1 Регистры защиты (PUx)

Каждый регистр защиты позволяет описать определенную область памяти и способы доступа к ней. Работа регистра защиты включается соответствующим битом в регистре управления. Подробное описание бит регистров защиты приведено ниже (Таблица 20).

**Таблица 20 – Регистр PU**

Бит	Имя	Назначение
10:0	STA	Начальный адрес модуля памяти. Соответствует физическому адресу памяти образуемому как {11'b0, START_A[10:0], 10'b0}
11	-	
22:12	ENDA	Конечный адрес модуля памяти. Соответствует физическому адресу памяти образуемому как {11'b0, END_A[10:0], 10'b11_1111_1111}
23	-	
25:24	JK_AP	Режим доступа к модулю со стороны шин J и K. 00 – полный доступ 01 – супервизор чтение и запись, пользователь чтение 10 – всем только чтение 11 – супервизор только чтение
27:26	I_AP	Режим доступа к модулю со стороны шины I. 00 – полный доступ 01 – только супервизор 10 – доступ запрещен 11 – доступ запрещен
29:28	H_AP	Режим доступа к модулю со стороны шины S (хост, DMA). 00 – чтение и запись 01 – только чтение 10 – доступ запрещен 11 – доступ запрещен
31:30	-	Резерв

Поля STA и ENDA используются для сравнения со значениями адресных шин J, K, I, S. При этом в сравнении участвуют только биты 20:10 указанных шин. Проверка выполняется только при доступе к внутренней памяти. Так для K шины

доступа попадание в некоторый модуль означает истинность следующего выражения:

$$\text{Hit\_PU} = (\text{K}[20:10] \geq \text{STA}) \ \&\& \ (\text{K}[20:10] \leq \text{ENDA}).$$

Минимальный размер модуля равен странице из 1 К слов. Модуль может рассматриваться как множество из 1 К страниц.

При нарушении прав доступа к модулю памяти вырабатывается исключительная ситуация.

### 5.10.2 Регистр управления (PU\_CR)

Регистр осуществляет включение в работу всех функций модуля защиты, а также функций управления кэш-памятью. После сброса значение регистра равно нулю. Подробное описание бит регистра управления приведено ниже (Таблица 21).

**Таблица 21 – Регистр PU\_CR.**

Бит	Имя	Назначение
0	DC_ON	1 – включение кэш памяти данных
1	IC_ON	1 – включение кэш памяти команд
2	EN_2dQW	1 – грузить 2 квадрослова данных при промахе 0 – загрузка одного квадрослова
3	EN_2iQW	1 – грузить 2 квадрослова команд при промахе 0 – загрузка одного квадрослова
4	SOC_speed_up	0 – разрешает выполнение команды на стадии W одновременно с загрузкой данных с СОК шины. 1 – запрещает выполнение команды на стадии W одновременно с загрузкой данных с СОК шины. Сначала загружаются данные, а затем начинается движение конвейера.
15:5	-	Зарезервировано. Биты общего назначения
16	PU0_EN	Разрешение работы модуля защиты 0: 1 – разрешено 0 – запрещено
17	PU1_EN	Разрешение работы модуля защиты 1: 1 – разрешено 0 – запрещено
18	PU2_EN	Разрешение работы модуля защиты 2: 1 – разрешено 0 – запрещено
19	PU3_EN	Разрешение работы модуля защиты 3: 1 – разрешено 0 – запрещено
20	PU4_EN	Разрешение работы модуля защиты 4: 1 – разрешено 0 – запрещено
21	PU5_EN	Разрешение работы модуля защиты 5: 1 – разрешено 0 – запрещено
22	PU6_EN	Разрешение работы модуля защиты 6: 1 – разрешено 0 – запрещено
23	PU7_EN	Разрешение работы модуля защиты 7: 1 – разрешено 0 – запрещено
31:24	-	Зарезервировано. Биты общего назначения

### **5.10.3 Регистр управления кэшированием данных**

Регистр используется для задания атрибута кэшируемости для отдельных страниц банков внешней памяти. Как известно процессор может адресовать три банка внешней памяти: MS0, MS1, SDRAM. При этом модули MS0, MS1 имеют деление на страницы по 128 К слов, а модуль SDRAM имеет деление на страницы по 512 К слов. Каждый бит регистра соответствует странице памяти. Так 0-й бит нулевой странице, 1-й первой и т. д. Итого можно определить до  $32 \cdot 128 = 4$  М слов памяти модулей MS0, MS1 и до 16 М слов динамической памяти.

### **5.10.4 Регистр управления сквозной записью**

Аналогично управлению кэшируемости страниц, имеется возможность для каждой страницы указать бит стратегии записи при попадании в кэш: 0 – запись только в кэш, 1 – запись в кэш и во внешнюю память.

### **5.10.5 Регистр управления кэшированием команд.**

Аналогично управлению кэшируемостью страниц для данных имеется возможность для каждой страницы указать бит кэшируемости при обращении за командами: 0 – не кэшируется, 1 – кэшируется.

## **5.11 Группы регистров отладки**

Группы отладки описываются ниже (Таблица 22). Доступ к регистрам отладки возможен только как к отдельным словам. К регистрам отладки может быть применена только команда пересылки «регистр-регистр» или командам немедленной загрузки данных. Эти регистры не могут быть загружены из памяти или сохранены напрямую в память. После каждой записи в регистры группы отладки происходит перезапуск конвейера процессора.

**Таблица 22 – Группа регистров отладки**

<b>Имя</b>	<b>Описание</b>	<b>Адрес</b>	<b>Значение по умолчанию</b>
WP0CTL	Управление точкой наблюдения 0	0x 0360	0x0000 0000
WP1CTL	Управление точкой наблюдения 1	0x 0361	0x0000 0000
WP2CTL	Управление точкой наблюдения 2	0x 0362	0x0000 0000
-	-	0x 0363	-
WP0STAT	Состояние точки наблюдения 0; (RO)	0x 0364	0x0000 0000
WP1STAT	Состояние точки наблюдения 1; (RO)	0x 0365	0x0000 0000
WP2STAT	Состояние точки наблюдения 2; (RO)	0x 0366	0x0000 0000
-	-	0x 0367	-
WP0L	Нижний адрес точки наблюдения 0	0x 0368	Не определено
WP0H	Верхний адрес точки наблюдения 0	0x 0369	Не определено
WP1L	Нижний адрес точки наблюдения 1	0x 036A	Не определено
WP1H	Верхний адрес точки наблюдения 1	0x 036B	Не определено
WP2L	Нижний адрес точки наблюдения 2	0x 036C	Не определено
WP2H	Верхний адрес точки наблюдения 2	0x 036D	Не определено
WPDR	Регистр данных точки наблюдения 1	0x 036E	0

WPMR	Регистр маски точки наблюдения 1.	0x 036F	0
CCNT0	Нижняя часть счетчика циклов	0x 0370	0
CCNT1	Верхняя часть счетчика циклов	0x 0371	0
PRFM	Маска монитора производительности	0x 0372	0x0000 0000
PRFCNT	Счетчик монитора производительности	0x 0373	0x0000 0000
TRCBMASK	Маска буфера трассировки (RO)	0x 0374	0x0000 0000
TRCBPTR	Указатель буфер трассировки (RO)	0x 0375	0x0000 0000
-	-	0x 0376– 0x 037B	-
TRCBVAL	Допустимый буфер трассировки (RO)	0x 037C	0x0000 0000
-	-	0x 037D– 0x 037F	-
TRCB31–0	Буфер трассировки (RO)	0x 0140– 0x015F	0x0000 0000

Точка наблюдения представляет собой механизм описания условия, при выполнении которого модуль отладки формирует запрос к процессору на обработку исключительной ситуации или на переход в режим эмуляции.

### 5.11.1 Регистр управления точкой наблюдения (WPxCTL)

У каждой из трех точек наблюдения есть регистр управления (WP0CTL, WP1CTL и WP2CTL), который используется для определения действий точки наблюдения. Значением сброса регистров WPxCTL является 0x0000 0000. Каждая из точек наблюдения осуществляет мониторинг определенных шин процессорного ядра: точка 0 – шину I, точка 1 – шины J и K, точка 2 – шину S.

Подробное описание разрядов регистра WP0CTL приведено ниже (Таблица 23).

**Таблица 23 – Регистр WP0CTL**

Бит	Имя	Назначение
1:0	OPMODE	Режим работы: 00 – выключена 01 – включена. Анализ совпадения адреса 10 – включена. Анализ попадания в интервал 11 – включена. Анализ выхода вне интервала
3:2	EXTYPE	Действия при выполнении условия: 00 – нет действий 01 – программное исключение 10 – переход в режим эмулятора 11 – резерв
4	SSTP	Режим пошагового выполнения команд: 1 – включен 0 – выключен
15:5	-	Всегда ноль
31:16	WP0CNT	Счетчик количества срабатываний точки наблюдения до генерации исключительной ситуации

Подробное описание разрядов регистра WP1CTL приведено ниже (Таблица 24).

**Таблица 24 – Регистр WP1CTL**

<b>Бит</b>	<b>Имя</b>	<b>Назначение</b>
1:0	OPMODE	Режим работы: 00 – выключена 01 – включена. Анализ совпадения адреса 10 – включена. Анализ попадания в интервал 11 – включена . Анализ выхода вне интервала
3:2	EXTYPE	Действия при выполнении условия: 00 – нет действий 01 – программное исключение 10 – переход в режим эмулятора 11 – резерв
4	READ	Мониторинг циклов чтения: 1 – включен 0 - выключен
5	WRITE	Мониторинг циклов записи: 1 – включен 0 – выключен
6	Jbus	Мониторинг шины J: 1 – включен 0 – выключен
7	Kbus	Мониторинг шины K: 1 – включен 0 – выключен
15:8	-	Всегда ноль
31:16	WP1CNT	Счетчик количества срабатываний точки наблюдения до генерации исключительной ситуации

Подробное описание разрядов регистра WP2CTL приведено ниже (Таблица 25).

**Таблица 25 – Регистр WP2CTL**

<b>Бит</b>	<b>Имя</b>	<b>Назначение</b>
1:0	OPMODE	Режим работы: 00 – выключена 01 – включена. Анализ совпадения адреса 10 – включена. Анализ попадания в интервал 11 – включена. Анализ выхода вне интервала
3:2	EXTYPE	Действия при выполнении условия: 00 – нет действий 01 – программное исключение 10 – переход в режим эмулятора 11 – резерв
4	READ	Мониторинг циклов чтения: 1 – включен 0 – выключен
5	WRITE	Мониторинг циклов записи: 1 – включен 0 – выключен
15:6	-	Всегда ноль
31:16	WP2CNT	Счетчик количества срабатываний точки наблюдения до генерации исключительной ситуации

### 5.11.2 Регистры состояния точки наблюдения (WPxSTAT)

У каждой из трех точек наблюдения есть регистр состояния (WP0STAT, WP1STAT и WP2STAT). Значениями сброса регистров WPxSTAT являются 0x0000 0000. Подробное описание разрядов регистра приведено ниже (Таблица 26).

**Таблица 26 – Регистр WPxSTAT**

Бит	Имя	Назначение
15:0	WPxCNT	Текущее значение счетчика точки наблюдения. Счетчик вычитает 1 каждый раз при срабатывании точки
17:16	EX	Состояние точки наблюдения: 00 – выключена 01 – активна 10 – резерв 11 – переполнился счетчик
31:18	-	Всегда 0

### 5.11.3 Регистры указателя адреса точки наблюдения (WPxL/WPxH)

Указатели адреса точки наблюдения являются 32-битными указателями, определяющими адрес или диапазон адресов точки наблюдения. После сброса их значение не определено.

### 5.11.4 Регистр маски монитора производительности (PRFM)

Регистр маски монитора производительности (PRFM) определяет, какие события в процессоре отслеживаются и подсчитываются счетчиком монитора производительности (PRFCNT). Значением сброса регистра PRFM является 0x0000 0000. Подробное описание разрядов регистра приведено ниже (Таблица 27).

**Таблица 27 – Регистр PRFM**

Бит	Имя	Назначение
7:0	-	Резерв
8	Jexe	Подсчет команд выполненных на линии конвейера J 1 – включен 0 – выключен
9	Kexe	Подсчет команд выполненных на линии конвейера K 1 – включен 0 – выключен
10	Xexe	Подсчет команд выполненных на линиях конвейера X1 и X2 1 – включен 0 – выключен
11	Yexe	Подсчет команд выполненных на линиях конвейера Y1 и Y2 1 – включен 0 – выключен
12	Sexe	Подсчет команд выполненных на линии конвейера S 1 – включен 0 – выключен
15:13	-	Резерв
16	STALL	Подсчет команд остановов конвейера вызванных ожиданием данных или другими блокирующими конвейер ситуациями 1 – включен 0 – выключен

17	BTB_true	Подсчет количества правильных предсказаний буфера переходов 1 – включен 0 – выключен
18	ABORT	Подсчет количества программных исключительных ситуаций 1 – включен 0 – выключен
19	Uexe	Подсчет количества линий команд выполненных в режиме пользователя 1 – включен 0 – выключен
20	BTB_false	Подсчет количества ошибочных предсказаний буфера переходов 1 – включен 0 – выключен
21	BTB_load	Подсчет количества загрузок в буфер переходов 1 – включен 0 – выключен
31:22	-	Резерв

Все события, которые отслеживаются монитором производительности, суммируются по «ИЛИ» и при их наступлении происходит увеличение счетчика PRFM на 1. Логичным является разрешение подсчета одного условия из многих.

#### **5.11.5 Регистр счетчика монитора производительности (PRFCNT)**

Назначение счетчика производительности – увеличивать свое значение на 1 каждый раз, когда происходит отслеживаемое событие. Отслеживаемое событие определяется регистром маски монитора производительности (PRFM). Регистр очищается после сброса.

#### **5.11.6 Регистры счетчика циклов (CCNTx)**

Длина счетчика циклов равна 64 разряда, однако доступен он как два универсальных 32-разрядных регистра – CCNT0 и CCNT1. После сброса значение счетчика равно нулю.

В связи с тем, что счетчик 64 бита, а доступ возможен только к частям регистра, необходимо соблюдение специальной процедуры чтения и записи регистра. При чтении сначала считывается нижняя часть (CCNT0), а затем верхняя часть (CCNT1). При чтении нижней части верхняя копируется в буфер и это гарантирует ее корректное значение. При записи сначала пишется нижняя часть (она попадает в буфер), а затем верхняя. При записи верхней полное 64-битное значение попадает в счетчик.

#### **5.11.7 Регистры указателя и буфера трассировки (TRCBx/TRCBPTR)**

Каждый раз, когда процессор выполняет переход на непоследовательно исполняемую команду, адрес непоследовательно выбранной команды (адрес перехода) записывается в один из регистров буфера трассировки. Первая запись осуществляется в буфер трассировки 0, вторая – в 1 и далее циклично. Указатель буфера трассировки определяет последний записанный буфер трассировки.

**Таблица 28 – Группа регистров буфера трассировки**

<b>Имя</b>	<b>Описание</b>	<b>Адрес</b>	<b>Значение по умолчанию</b>
TRCB0	Буфер трассировки 0; только чтение	0x 0140	0x0000 0000
TRCB1	Буфер трассировки 1; только чтение	0x 0141	0x0000 0000
TRCB2	Буфер трассировки 2; только чтение	0x 0142	0x0000 0000
TRCB3	Буфер трассировки 3; только чтение	0x 0143	0x0000 0000
TRCB4	Буфер трассировки 4; только чтение	0x 0144	0x0000 0000
TRCB5	Буфер трассировки 5; только чтение	0x 0145	0x0000 0000
TRCB6	Буфер трассировки 6; только чтение	0x 0146	0x0000 0000
TRCB7	Буфер трассировки 7; только чтение	0x 0147	0x0000 0000
TRCB8	Буфер трассировки 8; только чтение	0x 0148	0x0000 0000
TRCB9	Буфер трассировки 9; только чтение	0x 0149	0x0000 0000
TRCB10	Буфер трассировки 10; только чтение	0x 014A	0x0000 0000
TRCB11	Буфер трассировки 11; только чтение	0x 014B	0x0000 0000
TRCB12	Буфер трассировки 12; только чтение	0x 014C	0x0000 0000
TRCB13	Буфер трассировки 13; только чтение	0x 014D	0x0000 0000
TRCB14	Буфер трассировки 14; только чтение	0x 014E	0x0000 0000
TRCB15	Буфер трассировки 15; только чтение	0x 014F	0x0000 0000
TRCB16	Буфер трассировки 16; только чтение	0x 0150	0x0000 0000
TRCB17	Буфер трассировки 17; только чтение	0x 0151	0x0000 0000
TRCB18	Буфер трассировки 18; только чтение	0x 0152	0x0000 0000
TRCB19	Буфер трассировки 19; только чтение	0x 0153	0x0000 0000
TRCB20	Буфер трассировки 20; только чтение	0x 0154	0x0000 0000
TRCB21	Буфер трассировки 21; только чтение	0x 0155	0x0000 0000
TRCB22	Буфер трассировки 22; только чтение	0x 0156	0x0000 0000
TRCB23	Буфер трассировки 23; только чтение	0x 0157	0x0000 0000
TRCB24	Буфер трассировки 24; только чтение	0x 0158	0x0000 0000
TRCB25	Буфер трассировки 25; только чтение	0x 0159	0x0000 0000
TRCB26	Буфер трассировки 26; только чтение	0x 015A	0x0000 0000
TRCB27	Буфер трассировки 27; только чтение	0x 015B	0x0000 0000
TRCB28	Буфер трассировки 28; только чтение	0x 015C	0x0000 0000
TRCB29	Буфер трассировки 29; только чтение	0x 015D	0x0000 0000
TRCB30	Буфер трассировки 30; только чтение	0x 015E	0x0000 0000
TRCB31	Буфер трассировки 31; только чтение	0x 015F	0x0000 0000

### **5.11.8 Регистр данных точки наблюдения 1 (WPDR)**

Точка наблюдения 1 имеет дополнительные возможности по мониторингу обмена данными ядра процессора с памятью. Выше описывался механизм срабатывания исключительной ситуации в случае совпадения адресов точки наблюдения с адресами шин J и K. Однако имеется возможность дополнить сравнение адресов еще и сравнением данных. При чтении имеется возможность контролировать данные, которые считывает процессор. Если дополнительно к совпадению адресов происходит совпадение прочитанных данных – точка срабатывает. Также можно контролировать и процесс записи данных. Необходимые для контроля данные записываются в регистр WPDR. Необходимо отметить, что шина чтения или записи имеет разрядность 128 бит. Разрядность регистра данных только 32 бита. Поэтому сравнение выполняется только для 32-разрядного слова

шины данных определяемого младшими битами шины адреса независимо от того, какое действие выполняется: чтение или запись 64-х или 128-разрядного слова.

#### **5.11.9 Регистр маски точки наблюдения 1 (WPMR)**

При использовании шины данных в описании точки наблюдения 1 не все биты шины данных могут понадобиться при анализе читаемых или записываемых данных. Регистр маски позволяет установить, какие биты регистра данных WPDR должны участвовать в сравнении. Если бит установлен в 1 – соответствующий ему бит данных сравнивается с битом шины данных. После сброса регистр маски равен нулю и данные не участвуют в работе точки наблюдения.

## 6 Встроенный интерфейс (SOC-интерфейс)

При работе с внутренней памятью ядро процессора использует три внутренние шины I, J, K. Однако для работы с периферийными устройствами процессор использует специальный встроенный интерфейс, который осуществляет передачу запроса от внутренних шин ядра на специальную шину периферийных устройств (SOC – шина). Данная шина соединяет периферийные модули (внешний порт, DMA, порты связи, порт JTAG, контроллер прерываний и др.) с системой памяти через SOC-интерфейс и шину S типа. SOC шина имеет шину данных шириной 128 бит и адресную шину шириной 32 бита. Она работает с частотой равной половине частоты ядра. Все данные, передаваемые между внутренней памятью или ядром и периферийными модулями, проходят через SOC-интерфейс и SOC-шину.

SoC-интерфейс представляет собой модуль процессора, выполняющий связующую роль между ядром и внутренними периферийными устройствами. Со стороны ядра к интерфейсу подключены четыре шины J-, K-, I- и S-типов, а с другой стороны одна SoC-шина периферийных устройств. SoC-шина имеет пропускную способность 128 бит, и ее тактовый генератор (SOCCLK) работает на частоте, равной половине частоты ядра (CCLK). Периферийными устройствами, которые могут быть ведущими на SoC-шине являются (Рисунок 10):

- SoC – интерфейс;
- контроллер DMA.

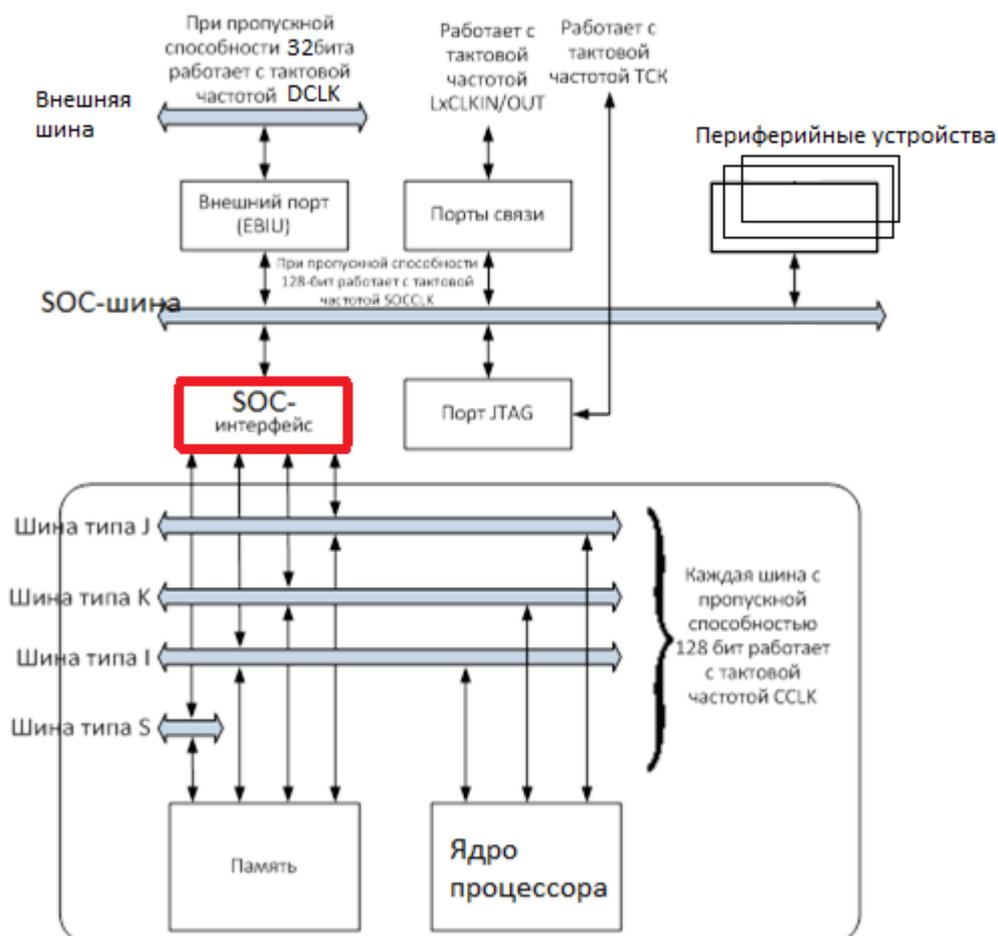


Рисунок 10 – Подключение периферийных модулей

Все внутренние периферийные устройства подключены к SOC-шине и являются ведомыми устройствами.

Структура SOC-интерфейса показана на Рисунок 11, а архитектура шин, соединяющих различные модули SOC-шины, приведена на Рисунок 12.

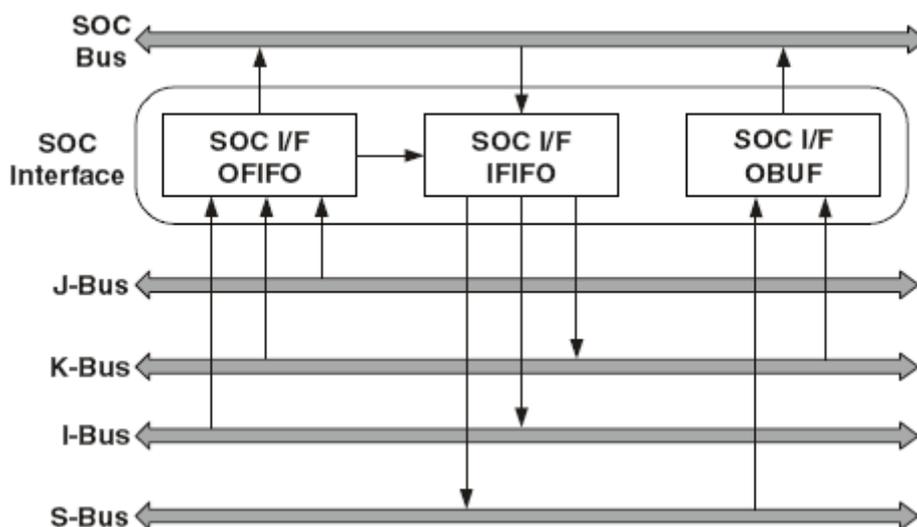


Рисунок 11 – Поток данных через SOC-интерфейс

В связи с тем, что на SOC-шине может быть несколько ведущих устройств, при каждом доступе к шине выполняется арбитраж. При этом доступ к периферийным устройствам возможен со стороны процессора и со стороны контроллера DMA. Со стороны процессора представителем является SOC-интерфейс и при этом он имеет два источника запросов: выходное FIFO интерфейса (OFIFO) и выходной буфер (OBUF). Приоритеты доступа при арбитраже фиксированы. Наивысший приоритет имеет выходной буфер интерфейса OBUF, который может возвращать прочитанное значение в контроллер DMA или в порт связи.

SOC-интерфейс состоит из трех устройств FIFO, через которые проходят все потоки данных. Если процессор обращается во внешнюю память, запрос из SOC-интерфейса поступает сразу во внешний интерфейс. Если шины процессорного ядра посылают в SOC-интерфейс запрос на чтение, прочитанные данные должны вернуться обратно в интерфейс. Для возвращаемых данных используется входное IFIFO. Также в данный буфер помещаются запросы всех внешних ведущих устройств, которые обращаются к внутренней памяти ядра. Если модуль IFIFO получил прочитанные данные, он пересылает их во внутренний регистр ядра. Если же был получен запрос на запись со стороны ведущего устройства, записываемые данные пересылаются по адресу-приемнику (используется шина S). В случае, если запрос на чтение получен от внешнего ведущего устройства, выполняется чтение внутренней памяти (используется шина S) и прочитанные данные помещаются в выходной буфер OBUF. Таким образом, в OBUF попадают только считываемые внешними ведущими устройствами данные ядра.

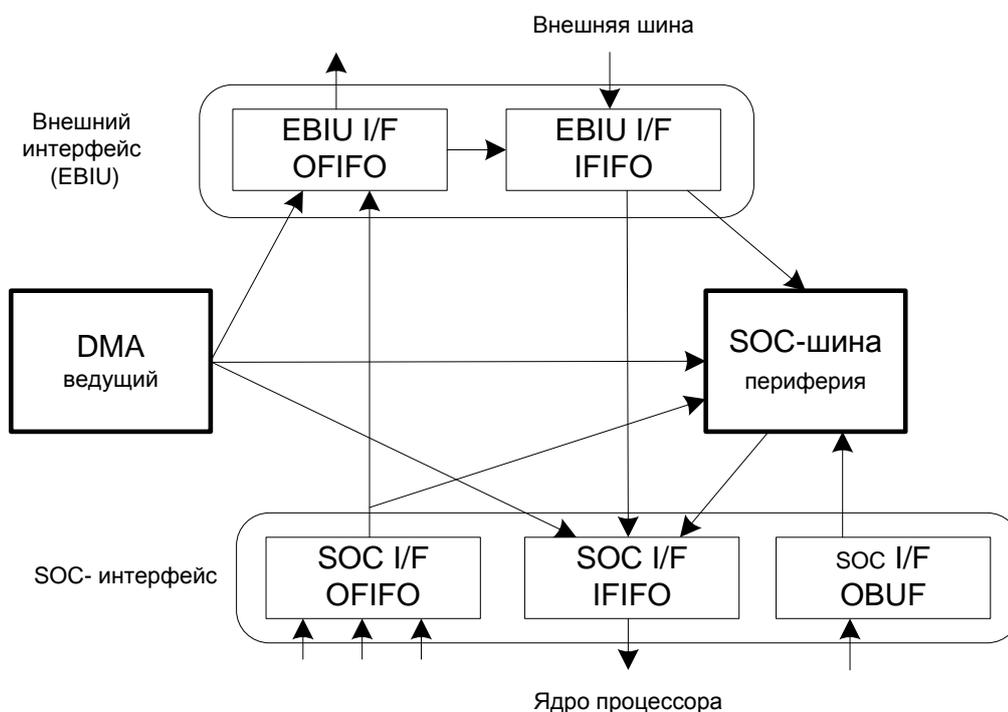


Рисунок 12 – Структура соединений модулей SOC-шины

## 6.1 Транзакции SOC OFIFO

Все запросы ядра по шинам J, K, I, которые не попадают во внутреннюю память, помещаются в выходное OFIFO. Это могут быть следующие запросы:

- Запись из ядра процессора по шинам J и K в устройства SOC-шины или внешнее адресное пространство;
- Запросы чтения по шинам J и K из устройств SOC-шины или внешнего адресного пространства;
- Запрос чтения команды по шине I из внешней памяти или из регистра инструкций эмулятора (EMUIR).

Все запросы ядра поступают в OFIFO с частотой ядра (CCLK). Считывание запросов из буфера выполняется с частотой SOC-шины (SOCCLK). На выходе буфера осуществляется проверка, к какому ресурсу выполняется запрос (внешняя память или внутренняя периферия), и далее запрос пересылается в пункт назначения. Если выполняется запрос к периферии, он имеет наименьший приоритет в арбитраже доступа. Если выполняется запрос к внешней памяти, также происходит арбитраж доступа к выходному FIFO внешнего интерфейса. За доступ к внешнему ресурсу OFIFO соревнуется с контроллером DMA (Рисунок 12). SOC-интерфейс в этом арбитраже всегда имеет наивысший приоритет.

С точки зрения конвейера процессора команда завершается, когда запрос на запись или на чтение помещается в SOC OFIFO. Если это операция записи, она считается выполненной и конвейер ядра продолжает движение, а если операция чтения, конвейер останавливается до момента возвращения данных. При этом неважно имеется зависимость по данным или нет. При чтении команд из внешней памяти после помещения двух запросов в OFIFO конвейер команд останавливается до момента получения прочитанной команды. При этом конвейер обработки команд может продолжать выбирать команды из буфера выравнивания и отправлять их на исполнение.

Буфер OFIFO имеет размер 8 записей. Это означает, что он может принять до 8-ми запросов без остановки конвейера процессора. При этом имеются следующие особенности:

- Если по шинам J и K одновременно выполняются две записи, процессору потребуется один такт, чтобы поместить их в OFIFO. При этом одна транзакция (J) помещается сразу в OFIFO, а вторая (K) во временный буфер. Из временного буфера запрос пересылается в OFIFO в следующем такте.
- Если процессору не требуется доступ к OFIFO в следующем такте, задержек конвейера не произойдет.

Запросы в OFIFO помещаются с частотой ядра, а считываются с частотой SOC-шины, т.е. в два раза меньшей. Поэтому в случае интенсивного потока записи во внешнюю память возможно заполнение буфера и это вызовет приостановку конвейера процессора.

## **6.2 Транзакции SOC IFIFO**

Входной буфер IFIFO SOC-интерфейса обрабатывает два типа транзакций:

- Запросы чтения ядра процессора. SOC IFIFO принимает прочитанные данные, полученные в результате транзакций чтения, которые были переданы через OFIFO;
- Запросы чтения или записи внутренней памяти ядра от DMA.

В первом случае возвращаемые данные записываются в регистр ядра, используя шину K. Во втором случае IFIFO использует шину S для доступа к внутренней памяти. В случае операции чтения внутреннего ресурса, прочитанные данные помещаются в выходной буфер OBUF.

Имеется три источника запросов к IFIFO (Рисунок 12):

- DMA;
- внешний интерфейс;
- SOC-шина.

Наивысший приоритет имеет SOC-шина, т.к. ее запрос означает чтение регистра периферии по запросу ядра, далее идет запрос внешнего интерфейса и наименьший приоритет у контроллера DMA. Запрос от внешнего интерфейса означает возвращение данных в ответ на запрос чтения внешней памяти ядром. Приоритеты доступа расставлены так, чтобы запросы ядра получали наивысший приоритет. В случае чтения это минимизирует простои процессора.

## **6.3 Транзакции SOC OBUF**

Выходной буфер OBUF SOC-интерфейса задействуется только в операциях чтения внутренних ресурсов (памяти) контроллером DMA. Данные из OBUF могут пересылаться в DMA или порты связи.

## **6.4 Программирование SOC интерфейса**

Чтение процессором внешней памяти вызывает передачу запроса от ядра через несколько доменов синхронизации. Все это может приводить к длительным остановкам процессора. Для ускорения работы процессора с внешней памятью

используется кэш-память данных и кэш-память команд. Процессор может организовывать работу с внешней памятью, как с использованием кэш-памяти, так и без.

Архитектурно SOC-шина представляет собой простой интерфейс к регистрам периферийных устройств, в котором циклы чтения или записи выполняются в течение одного такта. Если какое-либо из устройств не готово выдать (принять) данные в течение одного такта, оно может выставлять сигнал ожидания завершения операции.

## 7 Таймеры

Процессор имеет два 64-битных таймера общего назначения: Timer 0 и Timer 1. Таймеры автономные с загружаемым начальным значением и индикацией по его истечению.

Доступ к таймерам возможен как по номеру регистра (с использованием команд пересылки), так и при помощи адреса в адресном пространстве периферийных устройств (с использованием команд загрузки и сохранения). Таймеры принадлежат группе регистров контроллера прерываний (группа 0x3A). Номера регистров и адреса в адресном пространстве периферийных устройств приведены ниже (Таблица 29).

**Таблица 29 – Регистры модуля таймеров**

Имя	Описание	Номер/адрес	Значение по умолчанию
INTCTL	Управление прерыванием	0x074E / 0x8000_034E	0
TIMER0L	Текущее значение таймера 0 младшие разряды	0x 0750 / 0x8000_0350	Не определено
TIMER0H	Текущее значение таймера 0. старшие разряды	0x 0751/ 0x8000_0351	Не определено
TIMER1L	Текущее значение таймера 1. младшие разряды	0x 0752/ 0x8000_0352	Не определено
TIMER1H	Текущее значение таймера 1 старшие разряды	0x 0753 / 0x8000_0353	Не определено
TMRIN0L	Значение инициализации таймера 0. младшие разряды	0x 0754/ 0x8000_0354	Не определено
TMRIN0H	Значение инициализации таймера 0 старшие разряды	0x 0755/ 0x8000_0355	Не определено
TMRIN1L	Значение инициализации 1 младшие разряды	0x 0756 / 0x8000_0356	Не определено
TMRIN1H	Значение инициализации 1 старшие разряды	0x 0357 / 0x8000_0357	Не определено

Таймеры – это автономно работающие счетчики, имеющие следующие функции:

- загрузка начальным значением;
- декремент значение до нуля;
- индикация завершения счета;
- автоматическая перезагрузка;
- повторение цикла.

Индикатор завершения счета (такт после вычитания счетчика, когда он был равен 1) является прерыванием. Каждый таймер содержит два 64-битных регистра: начальное значение (TMRINxH/L) счета и текущее значение (TIMERxH/L) счетчика. Регистр TMRINxH/L доступен для чтения и записи, тогда как TIMERxH/L предназначен только для чтения. Оба регистра имеют размер 64 бита, но каждый доступен только как две 32-разрядные половины: старшая и младшая. Доступные 32-битные регистры:

- TMRIN0H;
- TMRIN0L;

- TMRIN1H;
- TMRIN1L;
- TIMER0H;
- TIMER0L;
- TIMER1H;
- TIMER1L.

При этом нет буферизации при считывании или записи различных частей регистра. Регистр TIMERx выполняет функцию счетчика и обновляется каждый такт частоты SOCCLK. Если счетчик в текущем такте имеет значение 1, в следующем такте в него загружается начальное значение из регистра TMRINx.

Каждый из таймеров имеет бит разрешения счета, который хранится в регистре управления INTCTL. Установка бита TMRxRN в 1 разрешает счет, а сброс бита в 0 останавливает счет.

## 7.1 Регистр управления прерываниями (INTCTL)

Регистр INTCTL – 32-битный регистр, который управляет чувствительностью внешних входов прерываний IRQ3-0 (прерываний чувствительных к фронту или уровню) и обеспечивает управление стартом и остановкой таймеров. Подробное описание разрядов регистра приведено ниже (Таблица 30).

**Таблица 30 – Регистр INTCTL**

Бит	Имя	Назначение
0	IRQ0_EDGE	Запрос прерывания по входу IRQ0 вызывается: 0 – фронтом сигнала (из высокого в низкий); 1 – уровнем (низкий)
1	IRQ1_EDGE	Запрос прерывания по входу IRQ1 вызывается: 0 – фронтом сигнала (из высокого в низкий); 1 – уровнем (низкий)
2	IRQ2_EDGE	Запрос прерывания по входу IRQ2 вызывается: 0 – фронтом сигнала (из высокого в низкий); 1 – уровнем (низкий)
3	IRQ3_EDGE	Запрос прерывания по входу IRQ3 вызывается: 0 – фронтом сигнала (из высокого в низкий); 1 – уровнем (низкий)
4	TMR0RN	Разрешение работы таймера 0: 0 – таймер остановлен; 1 – таймер работает
5	TMR1RN	Разрешение работы таймера 1: 0 – таймер остановлен; 1 – таймер работает
31:6	-	Не используются и при чтении всегда равны нулю.

## 7.2 Операции таймера

После сброса таймеры остановлены, т.к. биты TMRxRN в регистре INTCTL сбрасываются. Для запуска и работы таймера необходимо установить соответствующий бит в TMRxRN. После установки бита значение в TMRINxH/L копируется в регистр TIMERxH/L, и таймер начинает обратный отсчет.

Программы могут останавливать таймер, сбрасывая бит TMRxRN, и запускать его, устанавливая бит TMRxRN. Загрузка нового значения в регистр TMRINxH/L не

вызывает немедленной загрузки этого же значения в регистр счетчика TMRxH/L. Новое начальное значение будет загружено в счетчик только в двух ситуациях:

- при изменении бита TMRxRN из 0 в 1, т.е. при включении таймера;
- при работе таймера в момент достижения счетчиком значения 0.

При работающем таймере в момент достижения счетчиком нуля, таймер выполняет следующие действия:

- генерирует два прерывания с высоким и низким приоритетом;
- загружает в счетчик начальное значение;
- начинает обратный отсчет.

Таймеры имеют биты запроса прерывания в регистрах ILATx (INT\_TIMER0LP, INT\_TIMER1LP, INT\_TIMER0HP или INT\_TIMER1HP). Прерывание от таймера импульсное и после установки может быть сброшено при обслуживании данного запроса.

Для управления таймерами используйте следующую процедуру:

1. Если таймер работает, отключите таймер сбросом бита TMRxRN в регистре INTCTL;
2. Если прерывание должно произойти, установите высокий или низкий приоритет прерывания таймера, выполнив следующие действия:
  - назначьте адрес(а) вектора прерываний для программы, загрузив регистр(ы) вектора прерываний (IVTIMERxH/LP);
  - разрешите прерывания таймеров, установив бит INT\_TIMERxH/LP в регистре IMASK;
  - разрешите глобальные прерывания, установив бит SQCTL\_GIE в регистре SQCTL;
3. Загрузите начальное значение в регистры TMRINxH/L;
4. Запустите таймер, установив бит TMRxRN в регистре INTCTL.

## 8 Порты общего назначения GPIO

Большинство внешних выводов процессора представляют собой универсальные вход-выходы, которые могут быть перепрограммированы пользователем. Каждый бит портов GPIO программируется индивидуально.

Процессор имеет несколько портов ввода-вывода общего назначения. Часть из них имеет альтернативные функции ввода-вывода и может быть использована различными интерфейсами процессора. Процессор имеет 128 выводов общего назначения, которые могут быть индивидуально сконфигурированы как вход или выход. Также дополнительно имеется 22 вывода общего назначения, которые всегда сконфигурированы как выход.

### 8.1 Порты PA, PB, PC

Специальный модуль GPIO объединяет три самостоятельных 32-разрядных порта. Регистры этого модуля подключены к шине периферийных устройств и имеют базовый адрес 0x80001000 (порт A), адрес 0x80001040 (порт B), адрес 0x80001080 (порт C). Некоторые регистры имеют 4 адреса для работы: загрузки, установки, сброса и инверсии бит. Краткое описание регистров порта приведено ниже (Таблица 31) (порты A, B, C имеют идентичные регистры и индекс "I" в названии регистра подразумевает одно из трех значений: A, B, C).

Таблица 31 – Регистры управление портами PA, PB, PC

Номер	Имя	Тип	Назначение
0	PiDR_LD	R/W	Регистр данных для выдачи информации. Загрузка
1	PiDR_SET	R/W	Регистр данных для выдачи информации. Установка
2	PiDR_CLR	R/W	Регистр данных для выдачи информации. Сброс
3	PiDR_INV	R/W	Регистр данных для выдачи информации. Инверсия
4	PiDDR_LD	R/W	Регистр направления выдачи информации. 0 – прием; 1 – выдача Загрузка
5	PiDDR_SET	R/W	Регистр данных для выдачи информации. Установка
6	PiDDR_CLR	R/W	Регистр данных для выдачи информации. Сброс
7	PiDDR_INV	R/W	Регистр данных для выдачи информации. Инверсия
8	PIPEIE_LD	R/W	Регистр разрешения прерывания по положительному (из 0 в 1) изменению внешней линии порта. Загрузка
9	PIPEIE_SET	R/W	Регистр разрешения прерывания по положительному (из 0 в 1) изменению внешней линии порта. Установка
10	PIPEIE_CLR	R/W	Регистр разрешения прерывания по положительному (из 0 в 1) изменению внешней линии порта. Сброс
11	PIPEIE_INV	R/W	Регистр разрешения прерывания по положительному (из 0 в 1) изменению внешней линии порта. Инверсия
12	PINEIE_LD	R/W	Регистр разрешения прерывания по отрицательному (из 1 в 0) изменению внешней линии порта. Загрузка
13	PINEIE_SET	R/W	Регистр разрешения прерывания по отрицательному (из 1 в 0) изменению внешней линии порта. Установка
14	PINEIE_CLR	R/W	Регистр разрешения прерывания по отрицательному (из 1 в 0) изменению внешней линии порта. Сброс
15	PINEIE_INV	R/W	Регистр разрешения прерывания по отрицательному (из 1 в 0) изменению внешней линии порта. Инверсия

16	PiINVR_LD	R/W	Регистр инверсии входной информации внешней линии порта. Загрузка
17	PiINVR_SET	R/W	Регистр инверсии входной информации внешней линии порта. Установка
18	PiINVR_CLR	R/W	Регистр инверсии входной информации внешней линии порта. Сброс
19	PiINVR_INV	R/W	Регистр инверсии входной информации внешней линии порта. Инверсия
20	PiIMR_LD	R/W	Регистр маскирования прерывания внешней линии порта. Загрузка
21	PiIMR_SET	R/W	Регистр маскирования прерывания внешней линии порта. Установка
22	PiIMR_CLR	R/W	Регистр маскирования прерывания внешней линии порта. Сброс
23	PiIMR_INV	R/W	Регистр маскирования прерывания внешней линии порта. Инверсия
24	PiALT	R/W	Выбор функции порта
25	PiALT_SET	R/W	Выбор функции порта. Установка
26	PiALT_CLR	R/W	Выбор функции порта. Сброс
27	PiALT_INV	R/W	Выбор функции порта. Инверсия
28	PiPUR_LD	R/W	Регистр включения резистора «к единице» на внешней линии порта. Загрузка
29	PiPUR_SET	R/W	Регистр включения резистора «к единице» на внешней линии порта. Установка
30	PiPUR_CLR	R/W	Регистр включения резистора «к единице» на внешней линии порта. Сброс
31	PiPUR_INV	R/W	Регистр включения резистора «к единице» на внешней линии порта. Инверсия
32	PiPXD	R/O	Значение внешней линии порта
33	PiINTREQ	R/O	Запросы прерывания по входам порта
34			
35	PiECLR	W/O	Сброс запросов прерываний по фронту (запись 1 вызывает сброс)

Каждая линия 32-разрядного порта может быть сконфигурирована как выход или как вход, может сгенерировать прерывание по положительному или/и отрицательному фронту сигнала или по уровню (выбор уровня задается битом инверсии входа). Все запросы прерываний собираются в один общий запрос, который поступает в контроллер прерывания.

Каждый бит порта может иметь альтернативную функцию. Если бит регистра PiALT имеет нулевое значение, соответствующая линия порта работает под управлением регистров порта. Если бит равен 1, линия порта переходит под управление заданного устройства процессора. Возможные альтернативные функции портов приведены в таблице описания выводов.

## **8.2 Порты PxD и PxA**

Порты PxA и PxD предназначены для организации интерфейса к внешней памяти. Если нет необходимости подключать внешнюю память к данному интерфейсу, тогда шина адреса и шина данных могут быть использованы как порты общего назначения. При этом порт xA представляет собой 22-разрядный однонаправленный выходной порт, а порт xD – 32-разрядный двунаправленный порт общего назначения. Порт PxA имеет только регистр данных. Порт PxD имеет регистр данных и регистр направления. Регистры портов подключены к шине обмена

периферийных устройств и имеют базовый адрес 0x8000\_0090. Регистры портов представлены ниже (Таблица 32).

**Таблица 32 – Регистры управления портами PxA, PxD**

Номер	Имя	Описание
0	PXD_D_load	Регистр данных порта PxD. Загрузка всех 32 бит
1	PXD_D_set	Тот же регистр, но установка при записи 1 в бит
2	PXD_D_clear	Тот же регистр, но сброс при записи 1 в бит
3	PXD_DIR	Регистр направления порта PxD. 1 в бите указывает на выдачу данных, а 0 – на прием
4	PXA_load	Регистр данных порта PxA. Загрузка всех 22 бит
5	PXA_set	Тот же регистр, но установка при записи 1 в бит
6	PXA_clear	Тот же регистр, но сброс при записи 1 в бит
8	PX_ALT	Управление альтернативными функциями портов xA и xD
9	PXD_D_pin	Чтение внешних контактов порта PxD
10	PxD_PU	Управление резисторами порта xD

### 8.3 Регистр PX\_ALT

Данный регистр задает различные конфигурации для выводов портов PxA, PxD. Назначение бит регистра приведено внизу (Таблица 33).

**Таблица 33 – Регистр управления альтернативными функциями**

Бит	Имя	Тип	Назначение
0	PDB0	R/W	1 – порт PxD[7:0] находится под управлением контроллера внешней памяти. 0 – порт PxD[7:0] находится под управлением PxD_D[7:0] и PxD_DIR[7:0], т.е. порт общего назначения.
1	PDB1	R/W	Тоже для бит 15:8
2	PDB2	R/W	Тоже для бит 23:16
3	PDB3	R/W	Тоже для бит 31:24
4	PAB0	R/W	1 – порт PxA[7:0] находится под управлением внешнего контроллера внешней памяти. 0 – порт PxA[7:0] находится под управлением PxA_D[7:0], т.е. порт общего назначения.
5	PAB1	R/W	Тоже для бит 15:8
6	PAB2	R/W	Тоже для бит 21:16
7	PDXF0	R/W	Управление подключением периферийных устройств (Таблица 34)
8	PDXF1	R/W	Управление подключением периферийных устройств (Таблица 34)
31-9	-	-	Не используются

Конфигурация для разрядов с 0-го по 31-й порта PxD простая: если биты PDB[3:0] равны нулю, это порт общего назначения (каждый байт порта управляется своим битом), а если какой-то бит PDB[3:0] равен 1, соответствующие ему разряды порта выполняют альтернативную функцию. При этом для двух младших байт (биты с 0-го по 15-й) альтернативной функцией является только функция внешней шины данных для подключения внешней памяти. *Хотя биты PDB2 и PDB3 управляют различными данными, но их значение всегда должно быть одинаковыми, т.е. два нуля или две единицы.*

Функции для разрядов с 16-го по 31-й порта PxD более сложные и дополнительно зависят от битов 7 и 8 регистра PX\_ALT. Возможные конфигурации для старших 16 битов приведены ниже (Таблица 34).

Таблица 34 – Функции порта PxD[31:16] (биты PDB[3:2]=11b)

Номер	PX_ALT[7,8]	Функция
16-23	0x	Шина данных внешней памяти. Вход-выход. Биты с 16-го по 23-й
	10	NAND флэш-память. Шина данных. Вход-выход. Биты с 0-го по 7-й
	11	МКПД интерфейс. Входы – 16, 17, 21, 22 Выходы – 18, 19, 20, 23
24-30	0x	Шина данных внешней памяти. Вход-выход. Биты с 24-го по 30-й
	10	NAND флэш-память. Шина управления. Выходы.
	11	МКПД интерфейс. Входы – 26, 27, 28, 29, 30 (RTA[0-4]) Выходы – 24, 25
31	0x	Шина данных внешней памяти. Вход-выход. Бит 31-й
	1x	Вход. Для контроллера NAND флэш-памяти это вход готовности RnB. Для контроллера МКПД это дополнительный вход задания адреса RTA[5].

После сброса регистр альтернативных функций равен нулю и порты PxA, PxD являются портами общего назначения. При этом состояние портов следующее:

- порт PxD находится в отключенном состоянии (вход);
- значение порта PxA равно 0.

При необходимости использования внешней памяти необходимо записать в регистр PX\_ALT нужную конфигурацию (задать разрядность шины данных и шины адреса).

При необходимости подключения внешней NAND памяти или при использовании интерфейса МКПД, также в регистре PX\_ALT необходимо задать соответствующую конфигурацию. При использовании данных контроллеров шина данных может иметь максимальную разрядность 16 бит.

Включение режима 16-разрядной внешней шины данных происходит посредством установки бита SYSCON[19] регистра управления. Установка данного бита оказывает влияние только на контроллер внешней шины, т.к. в случае 16-разрядной шины контроллер будет выполнять два последовательных чтения (или две записи) при обращении к внешней памяти. При этом конфигурация шины данных и шины адреса не зависит от бита SYSCON[19]. Можно, например, использовать только восемь младших разрядов шины данных и работать с ними как с 32-разрядной шиной.

## 8.4 Регистр FLAGREG

В архитектуре вычислительного ядра процессора предусмотрен специальный регистр управления флагами FLAGREG, который осуществляет управление специальными внешними контактами. Если для бит с 0-го по 3-й порта PC задать альтернативную функцию, выводы PC[3:0] переходят под управление регистра FLAGREG и начинают выполнять функции флагов FLAG3–0.

Каждый контакт может быть сконфигурирован как вход или выход индивидуально. При конфигурации как выход, программы могут использовать выводы флагов для сигналов событий или условий для любого внешнего устройства. При конфигурации как входы, программы могут считывать состояние внешних контактов, используя регистр FLAGREG или использовать значение входа как условие в условных командах.

Для подготовки и использования флагов следует использовать следующую процедуру:

- 1 С помощью битов FLAGx\_EN регистра FLAGREG сконфигурируйте каждый из выводов FLAG3–0 как вход (FLAGx\_EN=0) или выход (FLAGx\_EN=1).
- 2 С помощью битов FLAGx\_OUT регистра FLAGREG выберите значения выхода для каждого вывода FLAG3–0 (для тех контактов которые сконфигурированы как выход).
- 3 Установите в единицу биты с 0-го по 3-й в регистре альтернативных функций порта PC. С этого момента выводы PC[3:0] переходят под управление регистра флагов.
- 4 Прочитайте биты FLGx в регистре SQSTAT для определения входного значения для каждого вывода FLAG3–0, или используйте входные значения флагов (FLAGx\_IN) в условных командах.

Выводы флагов (FLAG3–0) позволяют процессору посылать сигналы внешним устройствам или получать входные сигналы от них. Отметим, что аналогичные функции по управлению внешними флагами можно реализовать и при помощи регистров порта PC. Отличие будет только в том, что управление с помощью регистра FLAGREG будет происходить быстрее, т.к. регистр FLAGREG относится к группе регистров устройства управления и тактируется клоком ядра CCLK.

**Таблица 34 – Альтернативные функции портов PxA, PxD**

Обозначение				Тип	Выполняемая функция
Функция 1	Функция 2	Функция 3	GPIO		
ХТО			PxA[22]	О	Синхронизация, выход для подключения резонатора
A[21]			PxA[21]	О	Шина адреса внешнего интерфейса
A[20]			PxA[20]	О	Шина адреса внешнего интерфейса
A[19]			PxA[19]	О	Шина адреса внешнего интерфейса
A[18]			PxA[18]	О	Шина адреса внешнего интерфейса
A[17]			PxA[17]	О	Шина адреса внешнего интерфейса
A[16]			PxA[16]	О	Шина адреса внешнего интерфейса
A[15]			PxA[15]	О	Шина адреса внешнего интерфейса
A[14]			PxA[14]	О	Шина адреса внешнего интерфейса
A[13]			PxA[13]	О	Шина адреса внешнего интерфейса
A[12]			PxA[12]	О	Шина адреса внешнего интерфейса
A[11]			PxA[11]	О	Шина адреса внешнего интерфейса
A[10]			PxA[10]	О	Шина адреса внешнего интерфейса
A[9]			PxA[9]	О	Шина адреса внешнего интерфейса
A[8]			PxA[8]	О	Шина адреса внешнего интерфейса
A[7]			PxA[7]	О	Шина адреса внешнего интерфейса
A[6]			PxA[6]	О	Шина адреса внешнего интерфейса
A[5]			PxA[5]	О	Шина адреса внешнего интерфейса
A[4]			PxA[4]	О	Шина адреса внешнего интерфейса
A[3]			PxA[3]	О	Шина адреса внешнего интерфейса
A[2]			PxA[2]	О	Шина адреса внешнего интерфейса

Обозначение				Тип	Выполняемая функция
A[1]			PXA[1]	O	Шина адреса внешнего интерфейса
A[0]			PXA[0]	O	Шина адреса внешнего интерфейса
DATA[31]	NF_nRDY	MIL2_RTAP	PXD[31]	I/O	Шина данных внешнего интерфейса
DATA[30]	NF_CS[2]	MIL2_RT4	PXD[30]	I/O	Шина данных внешнего интерфейса
DATA[29]	NF_CS[1]	MIL2_RT3	PXD[29]	I/O	Шина данных внешнего интерфейса
DATA[28]	NF_CS[0]	MIL2_RT2	PXD[28]	I/O	Шина данных внешнего интерфейса
DATA[27]	NF_CLE	MIL2_RT1	PXD[27]	I/O	Шина данных внешнего интерфейса
DATA[26]	NF_ALE	MIL2_RT0	PXD[26]	I/O	Шина данных внешнего интерфейса
DATA[25]	NF_RE	MIL2_OU2X	PXD[25]	I/O	Шина данных внешнего интерфейса
DATA[24]	NF_WE	MIL2_OU2N	PXD[24]	I/O	Шина данных внешнего интерфейса
DATA[23]	NF_D[7]	MIL2_OU2P	PXD[23]	I/O	Шина данных внешнего интерфейса
DATA[22]	NF_D[6]	MIL2_IN2N	PXD[22]	I/O	Шина данных внешнего интерфейса
DATA[21]	NF_D[5]	MIL2_IN2P	PXD[21]	I/O	Шина данных внешнего интерфейса
DATA[20]	NF_D[4]	MIL2_OU1X	PXD[20]	I/O	Шина данных внешнего интерфейса
DATA[19]	NF_D[3]	MIL2_OU1N	PXD[19]	I/O	Шина данных внешнего интерфейса
DATA[18]	NF_D[2]	MIL2_OU1P	PXD[18]	I/O	Шина данных внешнего интерфейса
DATA[17]	NF_D[1]	MIL2_IN1N	PXD[17]	I/O	Шина данных внешнего интерфейса
DATA[16]	NF_D[0]	MIL2_IN1P	PXD[16]	I/O	Шина данных внешнего интерфейса
DATA[15]			PXD[15]	I/O	Шина данных внешнего интерфейса
DATA[14]			PXD[14]	I/O	Шина данных внешнего интерфейса
DATA[13]			PXD[13]	I/O	Шина данных внешнего интерфейса
DATA[12]			PXD[12]	I/O	Шина данных внешнего интерфейса
DATA[11]			PXD[11]	I/O	Шина данных внешнего интерфейса
DATA[10]			PXD[10]	I/O	Шина данных внешнего интерфейса
DATA[9]			PXD[9]	I/O	Шина данных внешнего интерфейса
DATA[8]			PXD[8]	I/O	Шина данных внешнего интерфейса
DATA[7]			PXD[7]	I/O	Шина данных внешнего интерфейса
DATA[6]			PXD[6]	I/O	Шина данных внешнего интерфейса
DATA[5]			PXD[5]	I/O	Шина данных внешнего интерфейса
DATA[4]			PXD[4]	I/O	Шина данных внешнего интерфейса
DATA[3]			PXD[3]	I/O	Шина данных внешнего интерфейса
DATA[2]			PXD[2]	I/O	Шина данных внешнего интерфейса
DATA[1]			PXD[1]	I/O	Шина данных внешнего интерфейса
DATA[0]			PXD[0]	I/O	Шина данных внешнего интерфейса

## 9 Прерывания

Процессор имеет контроллер прерываний, который принимает и обслуживает запросы от внешних портов и от внутренних периферийных устройств. Возможно задание векторных прерываний и программных прерываний, задаваемых пользователем. Прерывания могут быть вложенными. При обработке прерывания выполняется переход по вектору прерывания, значение которого хранится в таблице векторов прерываний.

Процессор поддерживает обработку прерываний, являющихся индикаторами различных внешних или внутренних событий. Прерывания делятся на программные и аппаратные.

Для каждого запроса прерывания существует:

- регистр в таблице векторов прерываний (группы регистров 0×38 и 0×39);
- бит в регистре защелки прерывания ILAT;
- бит в регистре маски прерывания IMASK;
- бит в регистре приоритета прерывания PMASK.

Номера битов в регистрах одинаковы и соответствуют номеру прерывания. Этот же номер имеет и регистр вектора прерывания в таблице векторов.

Программные прерывания вызываются специальными командами или определенными ситуациями, которые возникают при выполнении команд.

Аппаратные прерывания вызываются различными событиями, которые происходят в периферийных устройствах, или входным сигналом на выводах внешних прерываний.

Прерывание может быть вызвано аппаратной ошибкой в процессоре, таких как:

- Ошибка в работе контроллера DMA;
- Ошибка порта связи.

Прерывания могут быть чувствительны к фронту или уровню сигнала:

- *Прерывания чувствительные к фронту* фиксируются при их появлении установкой определенного флага в регистре ILAT. Флаги хранятся до тех пор, пока не будут обслужены или сброшены по команде. Если флаг не сбрасывается во время обработки, новое событие не обнаруживается и запрос игнорируется.
- *Прерывания чувствительные к уровню* должны поддерживать активный уровень запроса до начала обработки, в противном случае они не видны. Если запрос продолжается после обработки, он считается новым запросом прерывания.

Чувствительные к уровню прерывания обычно появляются как некоторый активный результат в определенном доступном регистре и аннулируются чтением этого регистра или записью в него неактивного значения. Чувствительные к фронту прерывания инициируются событием (например, истечением таймера).

Каждое из прерываний процессора имеет регистр вектора прерываний, который является адресом процедуры обслуживающей данное прерывание. Весь регистровый файл называется Таблица Вектора Прерываний (IVT). Каждый регистр таблицы является 32-битным для того, чтобы подключать процедуры обработки прерываний из внешней или внутренней памяти.

## 9.1 Группы регистров контроллера прерываний

Данные группы регистров относятся к векторам прерываний, управлению прерываниями и регистрам статуса (Таблица 35, Таблица 36, Таблица 37). Все регистры контроллера прерываний доступны только как однословные.

### 9.1.1 Группы регистров векторов прерываний

Данная группа регистров доступна как из поля команды пересылки, с использованием номера группы и номера регистра в группе, так и с помощью команд загрузки и сохранения по адресу в адресном пространстве периферийных устройств. Базовый адрес таблицы векторов прерываний в адресном пространстве периферийных устройств имеет значение 0x8000\_0300. Смещение вектора по отношению к базовому адресу, а также номер регистра с учетом его группы приведены ниже (Таблица 35).

**Таблица 35 – Группа А регистров таблицы вектора прерываний**

Имя	Описание	Номер	Смещение
IVKERNEL	Прерывание ядра VDK	0x 0300	0
IVGPIO	Прерывание от портов общего назначения PA, PB, PC	0x 0301	1
IVTIMER0LP	Низкий приоритет таймера #0	0x 0302	2
IVTIMER1LP	Низкий приоритет таймера #1	0x 0303	3
IVUART0	UART 0	0x 0304	4
IVUART1	UART 1	0x 0305	5
IVLINK0	порт связи #0	0x 0306	6
IVLINK1	порт связи #1	0x 0307	7
IVLINK2	порт связи #2	0x 0308	8
IVLINK3	порт связи #3	0x 0309	9
IVNANDC	NAND флэш контроллер	0x 030A	10
IVMIL0	МКПД 0	0x 030B	11
IVMIL1	МКПД 1	0x 030C	12
IVDIGC	Цифровой коррелятор	0x 030D	13
IVDMA0	Регистр DMA #0	0x 030E	14
IVDMA1	Регистр DMA #1	0x 030F	15
IVDMA2	Регистр DMA #2	0x 0310	16
IVDMA3	Регистр DMA #3	0x 0311	17
IVARINC_RX	Приемник ARINC	0x 0312	18
-	-	-	19
IVARINC_TX	Передачик ARINC	0x0314	20
-	-	-	21
IVDMA4	Регистр DMA #4	0x 0316	22
IVDMA5	Регистр DMA #5	0x 0317	23
IVDMA6	Регистр DMA #6	0x 0318	24
IVDMA7	Регистр DMA #7	0x 0319	25
-	-	0x 031A	26
-	-	0x 031B	27
-	-	0x 031C	28
IVDMA8	Регистр DMA #8	0x 031D	29
IVDMA9	Регистр DMA #9	0x 031E	30
IVDMA10	Регистр DMA #10	0x 031F	31

**Таблица 36 – Группа В регистров таблицы вектора прерываний**

<b>Имя</b>	<b>Описание</b>	<b>Номер</b>	<b>Смещение</b>
IVDMA11	Регистр DMA #11	0x 0320	32
IVADDA0	UP\DOWN модуль 0	0x 0321	33
IVADDA1	UP\DOWN модуль 1	0x 0322	34
IVADDA2	UP\DOWN модуль 2	0x 0323	35
IVADDA3	UP\DOWN модуль 3	0x 0324	36
IVDMA12	Регистр DMA #12	0x 0325	37
IVDMA13	Регистр DMA #13	0x 0326	38
IVLCD	LCD интерфейс	0x 0327	39
-	-	0x 0328	40
IVIRQ0	Вывод регистра IRQ0	0x 0329	41
IVIRQ1	Вывод регистра IRQ1	0x 032A	42
IVIRQ2	Вывод регистра IRQ2	0x 032B	43
IVIRQ3	Вывод регистра IRQ3	0x 032C	44
IVSPI	SPI	0x 032D	45
IVSSI0	SSI 0	0x 032E	46
IVSSI1	SSI 1	0x 032F	47
VIRPT	Векторный регистр VIRPT	0x 0330	48
IVVCAM	Видеокамера	0x 0331	49
IVBUSLK	Вектор блокировки шины	0x 0332	50
IVH264	H264 декодер	0x 0333	51
IVTIMER0HP	Высокий приоритет таймера 0	0x 0334	52
IVTIMER1HP	Высокий приоритет таймера 1	0x 0335	53
IVALARM	RTC ALARM	0x 0336	54
IVTIC	RTC TIC	0x 0337	55
IVDOG	RTC WATCHDOG	0x 0338	56
IVHW	Аппаратная ошибка	0x 0339	57
-	-	-	58-63

### **9.1.2 Группа регистров управления контроллером прерываний**

В задачу регистров управления прерывания входит прием запроса на прерывание, его обработка, генерация номера прерывания для передачи в процессор. Список регистров приведен ниже (Таблица 37). Регистры доступны как по номеру, так и по адресу в пространстве адресов периферийных устройств.

**Таблица 37 – Группа регистров управления прерываниями**

<b>Имя</b>	<b>Описание</b>	<b>Номер\Адрес</b>	<b>Значение после сброса</b>
ILATL	ILAT, младшие разряды	0x 0340 \ 0x8000_0340	0
ILATH	ILAT старшие разряды	0x 0341 \ 0x8000_0341	0
ILATSTL	ILAT младшие разряды, установка	0x 0342 \ 0x8000_0342	Не определено
ILATSTH	ILAT старшие разряды, установка	0x 0343\ 0x8000_0343	Не определено

ILATCLL	ILAT младшие разряды, сброс	0x 0344 \ 0x8000_0344	Не определено
ILATCLH	ILAT старшие разряды, сброс	0x 0345 \ 0x8000_0345	Не определено
PMASKL	PMASK младшие разряды	0x 0346 \ 0x8000_0346	0
PMASKH	PMASK старшие разряды	0x 0347 \ 0x8000_0347	0
IMASKL	IMASK младшие разряды	0x 0348 \ 0x8000_0348	0
IMASKH	IMASK старшие разряды	0x 0349 \ 0x8000_0349	0
-	-	-	-
INTCTL	Управление прерыванием	0x 034E \ 0x8000_034E	0

### 9.1.3 Регистр управления прерываниями (INTCTL)

Регистр INTCTL – 32-битный регистр, который управляет чувствительностью внешних входов прерываний IRQ3-0 (прерываний чувствительных к фронту или уровню) и обеспечивает управление стартом и остановкой таймеров. Подробное описание разрядов регистра приведено ниже (Таблица 38).

**Таблица 38 – Регистр INTCTL**

Бит	Имя	Назначение
0	IRQ0_EDGE	Запрос прерывания по входу IRQ0 вызывается: 0 – фронтом сигнала(из высокого в низкий); 1 – уровнем (низкий)
1	IRQ1_EDGE	Запрос прерывания по входу IRQ1 вызывается: 0 – фронтом сигнала(из высокого в низкий); 1 – уровнем (низкий)
2	IRQ2_EDGE	Запрос прерывания по входу IRQ2 вызывается: 0 – фронтом сигнала(из высокого в низкий); 1 – уровнем (низкий)
3	IRQ3_EDGE	Запрос прерывания по входу IRQ3 вызывается: 0 – фронтом сигнала(из высокого в низкий); 1 – уровнем (низкий)
4	TMR0RN	Разрешение работы таймера 0: 0 – таймер остановлен; 1 – таймер работает
5	TMR1RN	Разрешение работы таймера 1: 0 – таймер остановлен; 1 – таймер работает
31:6	-	Не используются и при чтении всегда равны нулю

Описание регистров таймера и способов работы с таймерами приведено в подразделе «Таймеры».

### 9.1.4 Регистры защелки прерываний (ILATL/ILATH)

Все запросы прерываний, которые поступают в контроллер прерываний, фиксируются (запоминаются) в регистре ILAT. Регистр ILAT – это 64-битный регистр доступный только для чтения как два 32-битных регистра ILATH и ILATL.

- ILATL – младшая часть ILAT (биты 31:0);
- ILATH – старшая часть ILAT (биты 63:32).

Каждый бит соответствует одному прерыванию и устанавливается при появлении этого прерывания. Порядок битов запросов прерываний устанавливается в соответствии с приоритетами прерываний: 0 – самый низкий приоритет.

Регистр ILAT имеет псевдонимы для упрощения процедур установки и сброса бит регистра. Для установки используются регистры ILATSTL или ILATSTH, а для сброса ILATCLL или ILATCLH. Регистры-псевдонимы доступны только по записи. При установке записываемое значение 1 приводит к установке бита, ноль не изменяет значение. При сбросе запись нуля вызывает сброс, а запись единицы не изменяет значение.

### **9.1.5 Регистры маскирования прерываний (IMASKL/IMASKH)**

Каждый бит запроса прерываний, зафиксированный в регистре ILAT, имеет соответствующий ему бит разрешения обслуживания прерывания (или бит маскирования прерывания). Значение 1 в бите маски разрешает прохождение соответствующего ему запроса прерывания в ILAT для дальнейшей обработки.

Регистр IMASK – 64-битный регистр, доступный как два 32-битных регистра IMASKL и IMASKH. После сброса значение IMASK всегда равно 0.

### **9.1.6 Регистры приоритетного маскирования прерываний (PMASKL/PMASKH)**

Регистр PMASK – 64-битный регистр, доступный как два 32-битных регистра PMASKL и PMASKH. Регистры доступны только для чтения и отображают состояние запросов прерываний, которые поступают на приоритетный шифратор контроллера прерываний.

Каждый бит регистра запроса прерываний соответствует определенному источнику прерываний и имеет соответствующий ему (с тем же номером) бит в регистре маскирования IMASK и бит в регистре PMASK. Назначения бит этих регистров приведены ниже (Таблица 39 и Таблица 40).

**Таблица 39 – Биты регистров ILATL, IMASKL, PMASKL**

<b>Бит</b>	<b>Имя</b>	<b>Описание</b>
0	INT_KERNEL	Прерывание ядра
1	INT_GPIO	Порты PA, PB, PC
2	INT_TIMER0LP	Низкий приоритет таймера #0
3	INT_TIMER1LP	Низкий приоритет таймера #1
4	INT_UART0	UART0
5	INT_UART1	UART1
6	INT_LINK0	порт связи #0
7	INT_LINK1	порт связи #1
8	INT_LINK2	порт связи #2
9	INT_LINK3	порт связи #3
10	INT_NANDC	NAND контроллер
11	INT_MIL0	МКПД 0
12	INT_MIL1	МКПД 1
13	INT_DIGC	Цифровой коррелятор
14	INT_DMA0	DMA #0
15	INT_DMA1	DMA #1
16	INT_DMA2	DMA #2

17	INT_DMA3	DMA #3
18	INT_ARINC_RX	Приемник ARINC
19	-	-
20	INT_ARINC_TX	Передатчик ARINC
21	-	-
22	INT_DMA4	DMA #4
23	INT_DMA5	DMA #5
24	INT_DMA6	DMA #6
25	INT_DMA7	DMA #7
26	-	-
27	-	-
28	-	-
29	INT_DMA8	DMA #8
30	INT_DMA9	DMA #9
31	INT_DMA10	DMA #10

**Таблица 40 – Биты регистров ILATH, IMASKH, PMASKH**

<b>Бит</b>	<b>Имя</b>	<b>Описание</b>
0	INT_DMA11	DMA #11
1	INT_ADDA0	Модуль UP\DOWN 0
2	INT_ADDA1	Модуль UP\DOWN 1
3	INT_ADDA2	Модуль UP\DOWN 2
4	INT_ADDA3	Модуль UP\DOWN 3
5	INT_DMA12	DMA #12
6	INT_DMA13	P DMA #13
7	INT_LCD	Контроллер LCD
8	-	-
9	INT_IRQ0	Вывод IRQ0
10	INT_IRQ1	Вывод IRQ1
11	INT_IRQ2	Вывод IRQ2
12	INT_IRQ3	Вывод IRQ3
13	INT_SPI	SPI
14	INT_SSI0	SSI 0
15	INT_SSI1	SSI 1
16	INT_VIRPT	Векторный регистр VIRPT
17	INT_VCAM	Контроллер видеокамеры
18	INT_BUSLOCK	Блокировка шины
19	INT_H264	Декодер H264
20	INT_TIMER0HP	Высокий приоритет таймера 0
21	INT_TIMER1HP	Высокий приоритет таймера 1
22	INT_ALARM	Прерывание ALARM от RTC
23	INT_TIC	TIC прерывание от RTC
24	INT_WDOG	Прерывание от сторожевого таймера
25	INT_HWERR	Аппаратная ошибка
31:26	-	-

## **9.2 Операции с прерываниями**

Операциями с прерываниями являются следующие виды операций:

- установка запросов прерываний;
- обработка прерываний или исключений;
- возврат из прерывания.

Контроллер прерываний включает следующие регистры установки и управления:

- Регистр управления прерываниями (INTCTL);
- Регистры маски прерываний (IMASK) – IMASKH и IMASKL;
- Регистры маски приоритета (PMASK) – PMASKH и PMASKL;
- Регистры защелки прерываний (ILAT) – ILATH и ILATL.

Наличие запроса прерывания отражается активным (равен 1) битом в регистре ILAT. Если соответствующий ему бит в регистре маски IMASK установлен, запрос прерывания поступает на дальнейшую обработку. Регистр приоритетов PMASK хранит запросы прерываний, которые процессор принял к обработке. Приоритетом прерывания является номер бита в регистре PMASK (от 0 до 63). Регистр приоритетов позволяет блокировать низкоприоритетные запросы прерываний до момента окончания обработки более высокоприоритетного. Если прерывание разрешено, и его приоритет выше текущего обрабатываемого прерывания (или нет обработки прерываний в данный момент), контроллер прерываний формирует активный уровень запроса прерывания к процессору, сопровождая его номером прерывания и адрес-вектором процедуры обслуживания данного прерывания. Если бит GIE (бит глобального разрешения прерываний) в регистре SQCTL установлен, процессор прерывает текущий ход программы и переходит к выполнению процедуры обработки прерывания. Как только прерывание начинает обслуживаться, соответствующий данному прерыванию бит в регистре PMASK устанавливается в 1. Данная установка запрещает прохождение запросов прерываний с приоритетом ниже обрабатываемого. При этом процедура обработки прерывания может разрешить обработку более высокоприоритетных прерываний. Как только обработка прерывания заканчивается, соответствующий ему бит в PMASK сбрасывается. Это происходит при выполнении команд RTI или RDS. При этом команда RTI означает завершение процедуры обработки, а команда RDS означает, что обработка текущего прерывания может быть прервана другим прерыванием, т.е. возможны вложенные прерывания.

Активный запрос прерывания вызывает установку соответствующего бита в регистре ILAT. Существует возможность программно установить бит в этом регистре, т.е. приложение может эмулировать прерывание посредством записи в регистр ILAT. Запись по адресам ILATSTL или ILATSTH обновляет регистр ILAT значением логического «ИЛИ» старого содержимого и нового записываемого. В результате, каждый установленный бит в записанных данных устанавливает соответствующий бит в регистре ILAT. Установка бита прерывания вызывает в процессоре предположение о возникновении соответствующего прерывания. Биты прерывания могут быть также сброшены записью в адреса сброса ILATCLL или ILATCLH. Такая запись применяет оператор AND к записываемым данным и старым данным регистра ILAT. В этом случае нулевой бит во входных данных сбрасывает соответствующий бит в ILAT, тогда как установленный бит остается неизменным. Таким образом, приложение может сбросить ждущее прерывание до его исполнения.

Операции сброса или установки бит в ILAT должны выполняться при следующих условиях:

- Недействительные (резервные) биты не могут быть установлены. При записи по адресам установки или сброса резервные биты должны быть нулями.
- Биты программного исключения и эмуляторного исключения не могут быть установлены через ILATST. Для того чтобы вызвать такие исключения следует использовать команды TRAP и EMUTRAP.

- Прерывания, которые запускаются уровнем, не должны изменяться.
- Записи в ILATL и ILATH не должны предприниматься (т.е. только сброс или установка).
- Прерывание должно быть сброшено в то время, когда оно запрещено, иначе оно может быть обслужено до того как будет сброшено.
- Адреса установки и сброса работают с одинарными словами.

### 9.3 Программа обработки аппаратного прерывания

Действия, выполняемые при обработке прерываний можно разделить на две группы:

- действия, выполняемые контроллером прерываний;
- действия, выполняемые процессором.

Рассмотрим последовательность обработки запроса прерывания на примере прерывания от таймера 0. Действия контроллера прерываний будут следующие.

- Когда таймер включен и его счетчик имеет в текущем такте SOCCLK значение 1, вырабатывается активный уровень запроса прерывания. Активный запрос существует только один такт, которого достаточно, чтобы в следующем такте биты 2 и 52 регистра ILAT установились в 1. Бит 2 соответствует запросу прерывания от таймера 0 с низким уровнем приоритета, а бит 52 – с высоким уровнем.
- При ожидании прерывания с высоким уровнем приоритета бит 52 в регистре IMASK должен быть установлен в 1. Логическое «И» между битом запроса прерывания и битом маски равно 1, следовательно прерывание может быть принято к обслуживанию.
- Далее выполняется анализ регистра PMASK. Если биты регистра с 52-й по 63-й равны нулю, в обработке нет прерываний равного или более высокого приоритета, чем данное. В противном случае придется подождать, пока не будет обслужено более высокоприоритетное прерывание.
- Выполнения описанных выше условий достаточно чтобы контроллер прерываний сформировал запрос прерывания к процессору. Кроме активного уровня запроса на линии, контроллер передает в процессор номер прерывания (52), а также извлекает из таблицы векторов прерываний значение регистра с номером 52 и передает его в процессор. Это значение является адресом процедуры обработки данного прерывания.

Когда запрос прерывания поступает в процессор (в устройство управления), первым действием процессора является проверка бита глобального разрешения прерываний GIE регистра управления SQCTL. Если бит равен 1 – прерывания разрешены и процессор выполняет следующие действия:

- Запоминает номер и адрес-вектор обработки пришедшего прерывания.
- Выполняет сброс всего конвейера чтения команд вместе с буфером IAB.
- Запускает чтение команд по адрес-вектору прерывания.
- Конвейер обработки команд продолжает и заканчивает обработку всех команд, которые находились на конвейере в момент сброса конвейера чтения команд.
- Когда последняя команда конвейера обработки завершает свое выполнение, процессор повторно проверяет бит разрешения прерываний GIE и, если прерывания по-прежнему разрешены, первая команда процедуры обработки прерывания поступает из буфера команд на конвейер обработки.

- При достижении первой командой последней стадии конвейера (W) происходит формирование сигнала о том, что данное прерывание взято на обработку. Этот факт приводит к установке бита 52 регистра PMASK и сброс бита 52 в регистре ILAT. Установленный бит в регистре PMASK запрещает все запросы прерывания уровнем ниже 53 (т.е. новое прерывание от таймера 0 также запрещено). Вместе с этим процессор запоминает в специальном регистре RETI адрес следующей команды, которая должна быть выполнена при выходе из прерывания. Дополнительно процессор устанавливает внутренний флаг EXE\_ISR, который запрещает все прерывания. Флаг EXE\_ISR указывает на то, что процессор находится в состоянии обработки прерывания. Флаг доступен для чтения как 20-й бит регистра SQSTAT.
- Переход на обработку прерывания вызывает установку режима супервизора (если до этого был режим пользователя).
- Сброс бита 52 в регистре ILAT означает, что бит стал доступен для приема нового запроса прерывания от таймера 0.
- Далее процессор выполняет все команды, которые соответствуют процедуре обслуживания данного прерывания. Для выхода из процедуры обработки используется команда RTI.
- Выполнение команды RTI приводит к сбросу бита 52 в регистре PMASK, что разрешает прохождение всех прерываний, которые ранее блокировались. Также данная команда вызывает сброс флага EXE\_ISR, что снимает блокировку прерываний и включает функции бита GIE. Процессор переходит к выполнению команд по адресу из регистра RETI.

Процедура, описанная выше, соответствует типичной ситуации обработки при отсутствии каких-либо аномалий. Однако существуют некоторые особенности:

- Если перед повторной проверкой разрешения прерываний (после полного освобождения конвейера обработки) оказывается, что прерывания запрещены (это может быть сделано командой сбрасывающей бит GIE), тогда все действия по обработке прерывания отменяются и конвейер перезапускается с адреса, следующего за адресом последней выполненной команды.
- Если запрос прерывания из контроллера пришел в процессор, но бит GIE равен нулю, прерывание будет отложено. В это время в контроллере прерываний может появиться запрос с более высоким приоритетом, чем данное. Контроллер прерываний обработает его и передаст в процессор номер и адрес-вектор нового прерывания. Запрос от таймера 0 в этом случае будет отложен до момента окончания обработки нового прерывания.
- Возможна ситуация, когда запрос пришел из контроллера, но в это же время процессор выполнял сброс бита запроса прерывания в регистре ILAT. Эта ситуация приведет к тому что запрос из контроллера прерываний будет стабильным в течение одного или нескольких тактов. Тем не менее, в такой ситуации запрос может быть обработан. Если снятие запроса произошло после или во время сброса конвейера чтения команд, это означает, что запрос будет обслужен. Для надежного выполнения подобных операций необходимо предварительно запрещать прерывания битом GIE, а затем производить сброс в контроллере прерываний.

Обычно процедура обработки прерывания начинается с сохранения контекста процессора, т.е. сохранения на стеке всех регистров процессора которые будут использованы программой обработки прерывания. Это необходимо для того, чтобы после обработки прерывания восстановить прежнее состояние процессора и дать

возможность прерванной программе корректно возобновить работу. Факт обработки прерывания отражается в установке флага EXE\_ISR и это запрещает все прерывание, в том числе и более высокоприоритетные, чем данное. Если время обработки прерывания значительно и это может повредить обработке запроса более приоритетного прерывания, программа может разрешить обслуживание более высокоприоритетных прерываний. Для этого необходимо обязательно сохранить на стеке содержимое регистра RETI, который хранит адрес возврата из прерывания. Для сохранения содержимого RETI используется специальный псевдоним этого регистра – RETIB.

Выполнение команды **j0 = RETIB;;** вызовет копирование регистра RETI в регистр j0 и сброс флага EXE\_ISR. Сам регистр j0 должен до этого быть сохранен на стеке. После выполнения указанной команды процессор имеет возможность обслуживать более высокоприоритетные запросы, чем запрос 52. Описываемая ситуация вносит коррективы в последовательность действий при выходе из обработки прерывания. Перед выходом мы имеем ситуацию, когда содержимое регистра возврата RETI неопределенно (новые прерывания могли изменить его). Поэтому необходимо взять наш адрес возврата из регистра j0 (или другого места, где он хранился) и вернуть обратно в RETI. Однако выполнение обычной пересылки **RETI = j0;;** будет некорректным, т.к. после этой команды возможно возникновение прерывания и адрес возврата может быть потерян. Снова следует использовать регистр-псевдоним RETIB. Пересылка **RETIB = j0;;** пересылает адрес возврата в RETI и устанавливает флаг EXE\_ISR. Это запрещает прерывания и позволяет безопасно восстановить весь контекст, а затем выполнить финишную команду RTI.

В выше описанной процедуре разрешения прерываний с более высоким приоритетом, чем текущее, разрешаются прерывания с приоритетом от 53 и выше, но прерывание с приоритетом 52 запрещено. Это означает, что обрабатывая запрос от таймера 0, имеется возможность принять новый запрос прерывания от таймера, т.к. бит 52 в регистре I LAT был сброшен и имеет возможность зафиксировать новое событие, но не обслуживать его (т.к. бит 52 в регистре PMASK установлен). Также нет возможности обслуживать запросы с приоритетом ниже 52.

Разрешить обслуживание запросов с приоритетом ниже 52 можно с помощью команды RDS. Выполнение команды **j0 = RETIB;;** разрешает только прерывания выше 52-го, т.к. бит 52 в регистре PMASK установлен. Выполнение команды RDS вызывает следующие действия процессора:

- сбрасывается в ноль текущий самый высокоприоритетный установленный бит в регистре PMASK;
- сбрасывается флаг EXE\_ISR;
- выполняется переход в режим работы, который предшествовал обработке прерывания;
- выполняется перезапуск конвейера в новых условиях.

Перед выполнением команды RDS необходимо сохранить адрес возврата. Если прежним режимом является режим пользователя с соответствующими ограничениями, и возврат в этот режим нежелателен, нужно сначала сохранить регистр SQCTL, и затем установить в нем бит режима супервизора. В этом случае после команды RDS останется режим супервизора.

Вышеописанные действия позволяют реализовать вложенные прерывания. Выход из подобных прерываний выполняется с использованием регистра RETIB.

Описанные действия относятся к обработке любого аппаратного прерывания, а не только 52-го.

Команда возврата из прерывания RTI является командой перехода и может выполняться с опцией предсказания или без. Использование опции предсказания позволяет ускорить выход из прерывания на несколько тактов процессора.

Необходимо помнить, что содержимое регистра RETI должно быть корректно в момент выполнения предсказания, т.е. в момент передачи в память адреса линии команд содержащей RTI. Если регистр RETI загружается непосредственно перед командой RTI или в нескольких тактах до RTI, предпочтительнее использовать команду RTI без предсказания перехода, т.к. в данной случае предсказание будет ошибочным, и все выполненные действия будут отменены.

## **9.4 Особенности сохранения контекста процессора**

Обычно сохранение контекста на стеке являются первыми действиями, которые выполняет программа обработки прерывания. В процессоре отсутствует специализированный указатель стека, т.к. любой регистр модуля J или K может выполнять эту функцию. Однако С-компилятор резервирует для указателей стека регистры J27 и K27, поэтому желательно использовать эти регистры в качестве указателя. В данном варианте процессора эти регистры могут быть дублированы для каждого из режимов (т.е. режим пользователя имеет свой указатель, а супервизор свой). Поскольку возможно использование сразу двух указателей стека, сохранение контекста может быть ускорено, если стеки размещаются в различных банках памяти. Это позволяет за такт сохранить восемь регистров процессора.

## **9.5 Обработка программных прерываний (исключений)**

Исключения – программные прерывания, которые вызваны исполняемым программным кодом. При этом могут быть специальные команды (TRAP, EMUTRAP) или ситуации, которые возникают при выполнении команд. К таким ситуациям относятся аномалии при вычислении с плавающей запятой, доступ к длинным словам по невыровненному адресу и др.

Обработка программного прерывания отличается от аппаратного следующим:

- Бит GIE не имеет значения при обработке исключений. Для программных прерываний используется специальный бит глобального разрешения SWIE.
- Часть программных прерываний имеет индивидуальные биты разрешения. Это касается исключений, которые возникают в вычислительных модулях X и Y. Другие исключения не имеют специальных бит для разрешения, кроме глобального.
- Программное прерывание происходит сразу за выполнением команды, инициировавшей это событие.
- Для хранения адреса возврата используется специальный регистр RETS, а не регистр RETI.
- Возникновение исключения приводит к установке 62-го бита регистра PMASK, что означает блокировку всех аппаратных прерываний.
- В качестве адреса-вектора процедуры обработки исключений используется содержимое регистра IVSW.
- Возникновение исключения приводит к установке специального бита EXE\_SWI. Бит виден в 21-м разряде регистра SQSTAT. Бит может быть установлен только при возникновении исключительной ситуации. Бит может быть сброшен только при выходе из обработки исключения или командой RDS. Других механизмов установки или сброса данного бита нет.
- Установка бита EXE\_SWI приводит к переходу в режим супервизора (если ранее был режим пользователя), а также к запрещению всех программных и аппаратных прерываний.

- Если при обработке исключительной ситуации возникло новое исключение, оно не будет обработано.

Последовательность действий процессора, выполняемых при возникновении исключительной ситуации, следующая:

- В поле SQSTAT[11:8] процессор сохраняет код исключительной ситуации. Если ситуация была вызвана командой TRAP, в поле SQSTAT[7:3] дополнительно сохраняется параметр данной команды. Все это позволяет идентифицировать источник исключения.
- Сбрасывается весь конвейер процессора и начинается чтение команд по адресу из регистра IVSW.
- В регистре RETS сохраняется адрес возврата, а именно адрес первой команды линии следущей за линией команд вызвавшей исключение.
- Бит PMASK[62] устанавливается в 1, запрещая все аппаратные прерывания.
- Устанавливается внутренний флаг EXE\_SWI, переводя процессор в режим супервизора и запрещая все аппаратные и программные прерывания.
- Процессором выполняются все команды процедуры обработки исключительной ситуации.
- Перед выходом из прерывания программа копирует содержимое регистра RETS в регистр RETI с помощью команды RETI = RETS;;. Регистр RETIB, в данном случае, не должен использоваться.
- Выход из обработчика осуществляется с помощью команды RTI.

При возврате из программного прерывания используется регистр RETI, поэтому последовательность выхода может быть такой:

```
[j31+temporary address] = RETI;;  
RETI = RETS;;  
RTI (NP); RETI = [j31+temporary address];;
```

Таким образом видно, что программное прерывание при своем возникновении создает очень строгие ограничения в поведении процессора, также как и аппаратное. Однако возникновение аппаратного прерывания не запрещает обработку программного прерывания, если оно возникнет при выполнении команд аппаратного обработчика. При обработке программного прерывания можно разрешить аппаратные прерывания, выполнив команду RDS. Выполнение команды RDS приведет к следующему:

- Произойдет сброс бита PMASK[62], т.е. все аппаратные прерывания в контроллере прерываний станут доступны.
- Произойдет сброс бита EXE\_SWI. Аппаратные прерывания станут управляться битом GIE, а программные – SWIE. Произойдет возврат к предыдущему режиму работы.

При использовании команды RDS необходимо заранее учесть влияние последующих событий на корректное исполнение программного обработчика.

Программное прерывание может быть вложенным в аппаратное, если биты EXE\_ISR и EXE\_SWI равны 1. В подобной ситуации команды RTI или RDS оказывают влияние только на бит EXE\_SWI, и их исполнение означает переход на уровень аппаратного обработчика.

## **9.6 Обработка прерываний эмулятора.**

Прерывания эмулятора – события, которые приводят к переходу процессора в режим эмуляции, когда все его действия управляются посредством команд через интерфейс JTAG. Подобное прерывание может быть вызвано:

- Командой EMUTRAP;
- Срабатыванием точки наблюдения;
- Аппаратным запросом из JTAG интерфейса.

Обработка данного типа прерываний имеет следующие особенности:

- Специальный бит глобального разрешения DBGEN. Биты GIE и SWIE не влияют.
- Прерывания от точек наблюдения и из JTAG имеют дополнительные биты разрешения.
- Прерывание происходит сразу за командой, которая покинула последнюю стадию конвейера W в момент возникновения запроса.
- Для хранения адреса возврата используется специальный регистр DBGE.
- В качестве адреса-вектора процедуры обработки используется фиксированный адрес 0x001F\_03A4.
- Возникновение исключения приводит к установке специального бита EMUL. Бит виден в 22-м разряде регистра SQSTAT. Бит может быть установлен только при возникновении данной исключительной ситуации. Бит может быть сброшен только при выходе из обработки исключения командой RTI. Других механизмов установки или сброса данного бита не предусмотрено.
- Установка бита EMUL приводит к переходу в режим эмулятора. Также это приводит к запрещению всех программных и аппаратных прерываний.

Последовательность действий процессора, выполняемых при возникновении прерывания эмулятора, следующая:

- В поле SQSTAT[15:12] процессор сохраняет код исключительной ситуации, что позволяет идентифицировать источник исключения.
- Сбрасывается весь конвейер процессора и начинается чтение команд по адресу 0x001F\_03A4. При этом изменения адреса не происходит. Данный адрес соответствует регистру команд интерфейса JTAG, т.е. процессор все время исполняет те команды, которые ему выставляет JTAG интерфейс.
- В регистре DBGE сохраняется адрес возврата.
- Устанавливается внутренний флаг EMUL, переводя процессор в режим эмулятора и запрещая все аппаратные и программные прерывания.
- Процессором выполняются все команды, которые предоставляет ему JTAG интерфейс.
- Перед выходом из прерывания программа копирует содержимое регистра DBGE в регистр RETI с помощью команды RETI = DBGE;; . Регистр RETIB, в данном случае, не должен использоваться.
- Выход из обработчика осуществляется с помощью команды RTI.

При возврате из программного прерывания используется регистр RETI, поэтому последовательность выхода может быть такой:

```
[j31+temporary address] = RETI;;  
RETI = DBGE;;  
RTI (NP); RETI = [j31+temporary address];;
```

Прерывание эмулятора накладывает серьезные ограничения на перечень команд, которые должны поставляться интерфейсом JTAG. Например, запрещены любые команды перехода и т.д. При переходе в режим эмулятора блокируются некоторые аппаратные ресурсы. Например, таймеры останавливаются и не меняют свое значение до момента выхода из режима эмуляции.

## 9.7 Источники прерываний процессора

**Таблица 41 – Таблица вектора прерываний**

Приоритет	Прерывание	Описание	Срабатывание
63 (самый высокий)	-	-	
62	SW	Программное исключение	-
61	-		
60	-		
59	-		
58	-		
57	HWERR	Аппаратная ошибка	По уровню
56	WDOG	RTC. Сторожевой таймер	По фронту
55	TIC	RTC. Интервальный таймер	По фронту
54	ALARM	RTC. Будильник	По фронту
53	TIMER1H	Высокий приоритет таймера 1	По фронту
52	TIMER0H	Высокий приоритет таймера 0	По фронту
51	H264	Декодер H264	По уровню
50	BUSLOCK	Блокировка шины	По фронту
49	VCAM	Контроллер видеокамеры	По уровню
48	VIRPT	Векторное прерывание	По фронту
47	SSI1	I2S_1	По уровню
46	SSI0	I2S_0	По уровню
45	SPI	SPI	По уровню
44	IRQ3	Вывод IRQ3	По фронту или по уровню
43	IRQ2	Вывод IRQ2	По фронту или по уровню
42	IRQ1	Вывод IRQ1	По фронту или по уровню
41	IRQ0	Вывод IRQ0	По фронту или по уровню
40	-	-	-
39	LCD	Контроллер ЖКИ	По уровню
38	DMA13	DMA канал 13	По фронту
37	DMA12	DMA канал 12	По фронту
36	ADDA3	UP\DOWN 3	По фронту
35	ADDA2	UP\DOWN 2	По фронту
34	ADDA1	UP\DOWN 1	По фронту
33	ADDA0	UP\DOWN 0	По фронту
32	DMA11	DMA канал 11	По фронту
31	DMA10	DMA канал 10	По фронту
30	DMA9	DMA канал 9	По фронту
29	DMA8	DMA канал 8	По фронту
28	-	-	-
27	-	-	-
26	-	-	-
25	DMA7	DMA канал 7	По фронту

24	DMA6	DMA канал 6	По фронту
23	DMA5	DMA канал 5	По фронту
22	DMA4	DMA канал 4	По фронту
21	-	-	-
20	ARINC_TX	ARINC передатчик	По уровню
19	-	-	-
18	ARINC_RX	ARINC приемник	По уровню
17	DMA3	DMA канал 3	По фронту
16	DMA2	DMA канал 2	По фронту
15	DMA1	DMA канал 1	По фронту
14	DMA0	DMA канал 0	По фронту
13	DIGC	Цифровой коррелятор	По фронту
12	MIL1	МКПД 1	По фронту
11	MIL0	МКПД 0	По фронту
10	NANDC	NAND флэш	По уровню
9	-	-	-
8	-	-	-
7	LINK1	Запрос порта связи 1	По уровню
6	LINK0	Запрос порта связи 0	По уровню
5	UART1	UART 1	По уровню
4	UART0	UART 0	По уровню
3	TIMER1L	Низкий приоритет таймера 1	По фронту
2	TIMER0L	Низкий приоритет таймера 0	По фронту
1	GPIO	Порты PA,PB,PC	По уровню
0 (самый низкий)	KERNEL	ядро	По фронту

### 9.7.1 Прерывания истечения таймера

Возможны четыре прерывания от таймеров: два для каждого таймера, одно – с высоким приоритетом и одно – с низким. Два приоритета для одного таймера дают возможность пользователю выбрать уровень приоритета.

Регистры вектора прерываний таймера: IVTIMER0HP, IVTIMER1HP, IVTIMER0LP и IVTIMER1LP. В случае прерывания от таймера, устанавливаются оба бита приоритетов (высокий и низкий) в ILAT (например, для таймера 0 это биты 2 и 52).

Сброс или установка бита запроса прерывания от таймера в регистре ILAT с помощью адресов ILATSTx или ILATCLx оказывает влияние сразу на оба бита таймера, т.е. нельзя разграничить запрос низкого от запроса высокого приоритета.

### 9.7.2 Прерывания на обслуживание порта связи

Существуют два канала портов связи, которые обычно работают с выделенными им каналами DMA. Регистры вектора обслуживания порта связи: IVLINK0, IVLINK1. Когда данные поступают в буфер приемника порта связи, и соответствующий ему канал DMA не инициализирован, порт связи формирует запрос прерывания на обслуживание.

### 9.7.3 Прерывания завершения работы каналов DMA

Каждый из 14-ти каналов DMA может генерировать прерывание. Регистры вектора: IVDMA0, IVDMA1, IVDMA2, IVDMA3, IVDMA4, IVDMA5, IVDMA6, IVDMA7, IVDMA8, IVDMA9, IVDMA10, IVDMA11, IVDMA12 и IVDMA13. Если бит разрешения прерывания в регистре управления передачей DMA установлен, канал DMA генерирует прерывание по завершении передачи блока.

### 9.7.4 Прерывания от внешних источников

Порты общего назначения PA, PB, PC могут формировать запрос на прерывание от каждого из 96-ти внешних контактов. Каждый запрос может быть запрограммирован индивидуально. Однако все запросы от портов общего назначения имеют один вектор прерывания IVGPIO. Существует четыре внешних прерывания общего назначения IRQ3–0, которые могут быть обработаны особым образом. Эти внешние входы соответствуют контактам 3-0 порта PC. Для них существуют отдельные вектора прерывания (IRQ3–0): IVIRQ0, IVIRQ1, IVIRQ2 и IVIRQ3. Когда активируется один из входов, запускается прерывание (если они разрешены). Тип прерывания по фронту или уровню программируется в регистре INTCTL.

### 9.7.5 Векторное прерывание

Векторное прерывание – прерывание общего назначения для использования другим мастером. Другое ведущее устройство или процессор могут записывать адрес в этот регистр. Запись вызывает установку запроса прерывания, а записанное значение в регистре – адрес обработки прерывания. Регистр вектора прерывания – VIRPT.

### 9.7.6 Прерывание от захвата шины

Прерывание захвата шины происходит, когда установлен бит захвата шины в регистре BUSLOCK, и процессор становится хозяином шины. Регистр вектора захвата шины – IVBUSLK. Данное прерывание сохранено для совместимости с процессором 1967ВЦ2Ф. В процессоре 1967ВЦ3Т внешняя шина всегда принадлежит процессору, поэтому выполнение процедуры захвата шины не требуется. Однако установка бита BUSLOCK приведет к генерации прерывания.

### 9.7.7 Прерывание от аппаратной ошибки

Прерывание аппаратной ошибки указывает на одну из возможных ошибок:

- Ошибка в DMA. При инициализации некоторого канала обнаружена ошибочная ситуация.
- Широковещательное чтение из внешней памяти (аналогично 1967ВЦ2Ф). Бит 16 в SYSTAT активен, если произошла данная ошибка.
- Доступ к отключенной SDRAM. Бит 18 в SYSTAT активен, если произошло данное событие.
- Обращение к несуществующему адресному пространству. Бит 19 в SYSTAT активен, если произошло данное событие.

- Ошибка порта связи. В одном из регистров LSTATx в полях RER или TER активна ошибка.

Регистр вектора прерываний аппаратных ошибок – IVHW.

### **9.7.8 Прерывания RTC**

Модуль часов реального времени RTC может формировать три запроса прерывания. Прерывание ALARM происходит при совпадении счетчика времени с заданным значением, что позволяет реализовать функции будильника. Прерывание TIC является периодическим и формируется через заданный интервал времени. Данное прерывание может быть полезно при работе ОС. Прерывание сторожевого таймера формируется в момент, когда счетчик сторожевого таймера принимает значение 1. Это служит напоминанием системе о необходимости выполнить повторную инициализацию сторожевого таймера. Если повторная инициализация не выполняется в течение определенного времени, сторожевой таймер инициирует сброс всей системы.

### **9.7.9 Прерывания от периферийных устройств**

В процессоре имеется ряд периферийных устройств, которые могут формировать запросы прерываний к процессору. Подробное описание порядка формирования и обслуживания прерывания приведено в главах, посвященных каждому из периферийных устройств.

### **9.7.10 Программные исключения**

Программные исключения – прерывания, появляющиеся во время выполнения программного кода. Если происходит исключение и бит SWIE равен нулю – исключение теряется. Вектор прерываний программных исключений – IVSW.

Тип исключительной ситуации можно определить по номеру в регистре состояния процессора устройства управления. Соответствие номера типу исключительной ситуации приведено ниже (Таблица 42). Используя номер исключительной ситуации в регистре состояния, можно выполнить соответствующую обработку возникшей исключительной ситуации. При этом может понадобиться дополнительная информация, доступная из других источников или полученная в результате анализа кода командной строки, вызвавшей исключение.

**Таблица 42 – Программные исключения**

<b>Номер</b>	<b>Имя</b>	<b>Описание события</b>
0	trap	Вызывается выполнением команды TRAP. Дополнительный номер исключения можно получить из регистра состояния
1	Wp_abort	Срабатывание точки наблюдения в модуле отладки
2	I0	Исключительная ситуация в модулях XU
3	UNDEF	Неопределенная команда или сочетание команд
4	I1	Невыровненный адрес при доступе к длинным словам
5	I3+prot	Нарушение прав доступа к памяти
6	PRFMC	Переполнение счетчика событий
7	I2	Чтение из области 0x0C00_0000 – 0x0FFF_FFFF
8		
9	H264	Ошибочная ситуация в модуле H264

10		
11		
12		
13		
14		
15		

## 10 Прямой доступ к памяти

Встроенный контроллер DMA осуществляет передачу данных без участия процессора. Данные и код могут быть загружены в процессор через DMA-передачу, которая может быть осуществлена между:

- внутренней памятью процессора и внешней памятью, внешней периферией, внутренней памятью;
- внешней памятью и внешними периферийными устройствами;
- внешней памятью и портами связи или между двумя портами связи.

DMA контроллер имеет 13 каналов. Четыре канала позволяют выполнять универсальные пересылки типа «память-память». Восемь каналов выполняют пересылки типа «память-периферийное устройство». При этом четыре канала работают с устройствами-приемниками, а другие четыре – с устройствами-передатчиками. Один канал (12-й) является специализированным и может работать только с модулем цифрового коррелятора.

Прямой доступ к памяти (direct memory access, DMA) – механизм передачи данных без исполнения команд в ядре процессора. Встроенный контроллер DMA избавляет процессор от необходимости передачи данных между внутренней памятью и внешними устройствами\памятью или между внутренними периферийными устройствами и внешней или внутренней памятью. Он позволяет ядру процессора указать каналу команду передачи данных и вернуться к нормальной работе, тогда как контроллер DMA выполнит передачу данных в фоновом режиме.

Модуль DMA может быть ведущим или ведомым на шине SOC. При доступе к регистрам DMA со стороны процессорного ядра, модуль DMA выступает в роли ведомого устройства. При работе канала DMA по пересылке данных модуль DMA выступает в роли ведущего устройства и соревнуется в доступе к различным ресурсам с другими ведущими устройствами. Ресурсы, к которым посылает свои запросы контроллер DMA, приведены ниже (Рисунок 12). Это интерфейсы внутренней или внешней памяти, а также устройства SOC-шины.

В данном подразделе используются следующие термины:

- *Блок управления передачей (TCB)* – блок из четырех слов, который определяет набор параметров для DMA-операций.
- *Регистр TCB* – 128-разрядный регистр, который содержит TCB.
- *Загрузка цепочки TCB* – процесс, при котором контроллер DMA загружает новый TCB из памяти и автоматически инициализирует регистр TCB.
- *Транзакция канала DMA* – последовательность действий канала по пересылке одного операнда.
- *Операнд канала* – порция данных (одно, два или четыре слова), размер которой задается регистром управления и которой оперирует канал в одной транзакции.
- *Режим квитиования* – режим работы канала, в котором инициатором одной транзакции выступает запрос от устройства.
- *Выровненный адрес* – означает, что для двойного слова (64 бита) младший бит адреса всегда равен 0, а для квадрослова (128 бит) два младших разряда адреса должны быть равны нулю.

## 10.1 Архитектура контроллера DMA

Контроллер DMA включает в себя 14 каналов, предназначенных для выполнения различных типов передачи.

- Каналы с 0 по 3 являются универсальными и позволяют задавать произвольный источник и приемник, которые имеют адрес в карте памяти процессора.
- Каналы с 4 по 7 жестко закреплены за внутренними периферийными устройствами, которые выполняют функции передатчиков данных (выдача данных на внешние контакты), т.е. источник данных этих каналов DMA программируемый (внутренняя или внешняя память), а приемник ограничен набором внутренних периферийных устройств. Поэтому каналы с 4 по 7 могут выполнять только пересылки память-устройство.
- Каналы с 8 по 11 жестко закреплены за внутренними периферийными устройствами, которые выполняют функции приемников данных (прием информации с внешних контактов), т.е. источник данных этих каналов ограничен набором внутренних периферийных устройств, а приемник данных программируемый (внутренняя или внешняя память). Таким образом, каналы с 8 по 11 могут выполнять только пересылки устройство-память.
- Канал 12 может быть использован совместно с цифровым коррелятором и позволяет организовать передачу данных от 16 каналов коррелятора во внутреннюю память процессора.
- Канал 13 используется внешним хост-устройством для доступа к внутренним ресурсам процессора. Внешнее ведущее устройство, используя последовательный синхронный интерфейс, может выполнять программирование регистров канала 13 DMA, задавая транзакции чтения или записи. Структура канала 13 отличается от структуры каналов 0-12 и приведена в подразделе «Хост-интерфейс».

Передачи DMA характеризуются направлением потока данных от источника к приемнику. Если в роли источников или приемников выступает память, они описываются регистром TCB. Универсальные каналы с 0-го по 3-й имеют два TCB-регистра для описания источника и приемника, каналы с 4-го по 12-й имеют – по одному регистру TCB. Рисунок 13 показывает блок-схему DMA-контроллера.

DMA поддерживает так называемые цепочки операций, когда одна запрограммированная в канале пересылка после завершения работы вызывает автоматическое чтение из внутренней памяти новой команды пересылки без участия процессора. Эта техника в первую очередь используется для одного DMA-канала, но в некоторых случаях может использоваться и для разных каналов.

Процессор также имеет четыре контакта внешнего запроса DMA ( $nDMAR3-0$ ), которые позволяют внешним устройствам ввода\вывода запрашивать DMA-сервис. В ответ на внешний запрос по входу  $nDMARi$ , соответствующий  $i$ -й канал контроллера DMA выполняет передачу в соответствии с его инициализацией. Для инициализации канала программа должна записать в блок управления передачей (TCB) командное слово.

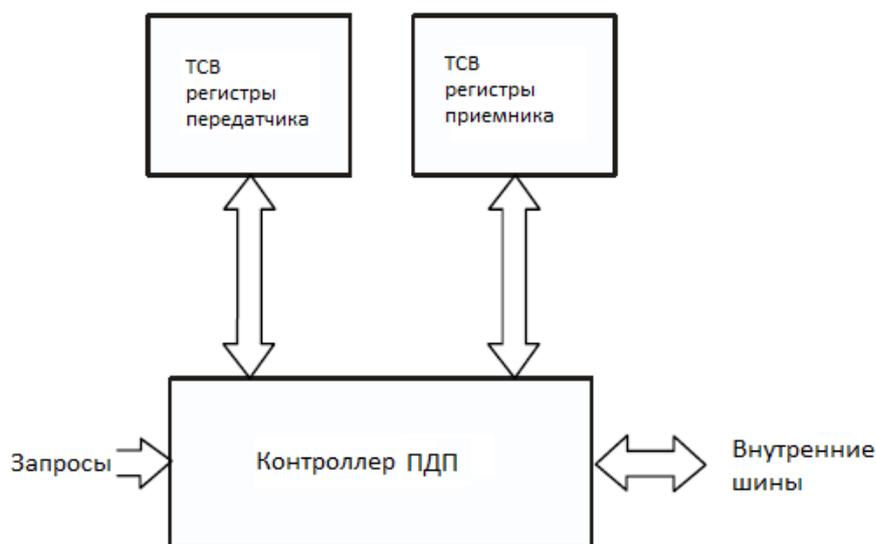


Рисунок 13 – Блок-схема DMA

Блок управления передачей ТСВ является регистром счетверенного слова (128 бит), который содержит информацию, необходимую для работы канала DMA. Вид регистра ТСВ представлен ниже (Рисунок 14). В одном канале может присутствовать приемник и передатчик, только приемник или только передатчик. Названия ТСВ передатчика и приемника имеют внутриканальный смысл: ТСВ передатчика передает данные ТСВ приемнику. Сам же ТСВ передатчик берет данные из источника информации, а ТСВ приемника передает принятые данные в пункт назначения.

В случае ТСВ передатчика, четыре слова содержат данные адреса источника информации (DI), количество слов для передачи (DX/DY), приращение адреса (DX/DY), контрольные разряды (DP).

В случае ТСВ приемника эти четыре слова содержат адрес назначения (DI), количество слов, которое нужно получить (DX/DY), приращение адреса (DX/DY), контрольные разряды (DP).



Рисунок 14 – Регистры блока управления передачей DMA

Одна передача (пересылка или транзакция) при работе DMA означает копирование порции данных (операнда) из источника данных в приемник данных. Размер порции данных описывается в регистре управления DP и может быть равен одному, двум или четырем 32-разрядным словам. В регистре DX задается количество слов для передачи. Количество транзакций будет равно количеству слов, деленному на размер порции данных. После каждой транзакции канал осуществляет изменение адреса на величину приращения, задаваемую в регистре DX(DY). Канал DMA может поддерживать двумерные пересылки данных, когда адрес изменяется по горизонтальной составляющей X и по вертикальной составляющей Y. Поэтому в ТСВ присутствуют описатели DX и DY. Обычный режим передачи – одномерный, используется только регистр DX. Регистр DY используется только в двумерном режиме.

Контроллер поддерживает словную адресацию памяти, и минимальной адресуемой единицей данных для него является 32-разрядное слово.

Соответствие между номером канала и его характеристиками приведено ниже (Таблица 43).

**Таблица 43 – Особенности программирования каналов контроллера**

<b>Номер канала</b>	<b>Источник данных</b>	<b>Приемник данных</b>	<b>Инициирование транзакции</b>
0	Программируется в ТСВ передатчика	Программируется в ТСВ приемника	Программируется. Может использовать запрос устройства
1	Программируется в ТСВ передатчика	Программируется в ТСВ приемника	Программируется. Может использовать запрос устройства
2	Программируется в ТСВ передатчика	Программируется в ТСВ приемника	Программируется. Может использовать запрос устройства
3	Программируется в ТСВ передатчика	Программируется в ТСВ приемника	Программируется. Может использовать запрос устройства
4	Программируется в ТСВ передатчика	Периферийное устройство	Всегда только по запросу периферийного устройства
5	Программируется в ТСВ передатчика	Периферийное устройство	Всегда только по запросу периферийного устройства
6	Программируется в ТСВ передатчика	Периферийное устройство	Всегда только по запросу периферийного устройства
7	Программируется в ТСВ передатчика	Периферийное устройство	Всегда только по запросу периферийного устройства
8	Периферийное устройство	Программируется в ТСВ приемника	Всегда только по запросу периферийного устройства
9	Периферийное устройство	Программируется в ТСВ приемника	Всегда только по запросу периферийного устройства
10	Периферийное устройство	Программируется в ТСВ приемника	Всегда только по запросу периферийного устройства
11	Периферийное устройство	Программируется в ТСВ приемника	Всегда только по запросу периферийного устройства
12	Запись данных цифровым коррелятором в регистр канала	Программируется в ТСВ приемника	Всегда при наличии данных во внутреннем буфере
13	Запись данных внешним ведущим устройством в регистр канала	Программируется в ТСВ приемника	Всегда при наличии данных во внутреннем буфере

Каналы контроллер DMA не однотипны. Каналы 0-3 позволяют выполнять пересылки: память-память, память-устройство, устройство-память. Каналы 4-7 позволяют выполнять только пересылки память-устройство, т.е. работают только с устройствами-передатчиками. Каналы 8-11 позволяют выполнять только пересылки устройство-память, т.е. работают только с устройствами-приемниками. При необходимости работы с внутренними устройствами с помощью каналов 0-3 возникают дополнительные ограничения. Каналы 0 и 2 могут быть запрограммированы только для работы с устройствами-передатчиками, а каналы 1 и 3 только с устройствами-приемниками.

Контроллер DMA может быть ведомым на SOC-шине устройств, и его внутренние ресурсы могут быть доступны процессору посредством чтения и записи регистров DMA. Доступ к регистрам контроллер DMA может быть выполнен как с помощью команд пересылки процессора (используя номер регистра), так и с помощью команд загрузки и сохранения (используя адрес в карте памяти). Базовый

адрес регистров контроллера в карте памяти периферийных устройств равен 0x8000\_0000. В таблицах описания регистров будет указано смещение от базового адреса.

### 10.1.1 Регистры управления и статуса

Общее описание группы регистров управления и статуса приведено ниже (Таблица 44).

**Таблица 44 – Группа регистров управления контроллер DMA**

<b>Имя</b>	<b>Описание</b>	<b>Адрес-смещение</b>	<b>Значение по умолчанию</b>
DCNT	Управление контроллер DMA	0x 0060	0x0000 0000
-	-	0x 0061–0x 0063	-
DCNTST	Управление контроллер DMA, биты установки	0x 0064	Не определено
-	-	0x 0065–0x 0067	-
DCNTCL	Управление контроллер DMA, биты сброса	0x 0068	Не определено
-	-	0x 0069–0x 006B	-
DSTAT	Статус контроллер DMA	0x 006C	0x0000 0000
-	-	0x 006D–0x 006F	-
DSTATCL	Статус контроллер DMA, биты сброса	0x 0070	Не определено
-	-	0x 0071–0x 0077	-
DMACFGL	Задание номера устройства Для каналов с 0 по 7	0x0078	0
DMACFGH	Задание номера устройства Для каналов с 8 по 15	0x0079	0
-	-	0x 007A–0x 007F	-

Особо отметим функции регистров DMACFGx (DMACFGL, DMACFGH). В процессоре имеется несколько периферийных устройств, способных поддерживать работу совместно с контроллер DMA. Каждому устройству в контроллер DMA ставится в соответствие номер. Для того чтобы некоторому каналу DMA поставить в соответствие выбранное устройство, в определенном поле регистра DMACFGx осуществляется программирование номера устройства. Для каждого канала в регистре конфигурации отведено четыре бита, что позволяет задать 16 источников запросов на обмен (биты 0-3 регистра DMACFGL соответствуют каналу 0, биты 4-7 каналу 1, биты 8-11 каналу 2 и т. д). Регистр DMACFGH обслуживает каналы с 8 по 11). После сброса значение регистра равно нулю и соответствует начальному варианту работы контроллер DMA. Значение в поле регистра соответствует номеру источника запроса.

Имеются две группы источников запросов: передатчики (Таблица 45) и приемники (Таблица 46).

Таблица 45 – Устройства-передатчики

Номер	Обслуживаемый запрос	Размер данных для обмена
0	Источник запроса определяется номером канала контроллер DMA: 0 – nDMAR0 1 – nDMAR1 2 – nDMAR2 3 – nDMAR3 4 – LINK 0 5 – LINK 1 6 – 7 – 8 – LINK 0 9 – LINK 1 10 – 11 –	-
1	Готовность передатчика UART_1 принять данные	32 бита
2	Готовность передатчика UART_2 принять данные	32 бита
3	Готовность передатчика SPI принять данные	32 бита
4	Готовность LCD контроллера принять данные	128 бит
5	Готовность передатчика SSI0 контроллера принять данные	32 бита
6	Готовность передатчика SSI1 контроллера принять данные	32 бита
7	Готовность буфера NANDF контроллера принять данные для записи	32 или 128 (определяется контроллером NANDF)
8	Готовность модуля ADDA0 принять данные	128 бит
9	Готовность модуля ADDA1 принять данные	128 бит
10	Готовность модуля ADDA2 принять данные	128 бит
11	Готовность модуля ADDA3 принять данные	128 бит
12	-	
13	-	
14	H264_RQ1	128 бит
15	H264_RQ0	128 бит

Таблица 46 – Устройства-приемники

Номер	Обслуживаемый запрос	Размер данных для обмена
0	Источник запроса определяется номером канала контроллер DMA: 0 – nDMAR0 1 – nDMAR1 2 – nDMAR2 3 – nDMAR3 4 – LINK 0 5 – LINK 1 6 – 7 – 8 – LINK 0 9 – LINK 1 10 – 11 –	

1	Готовность приемника UART_1 выдать данные	32 бита
2	Готовность приемника UART_2 выдать данные	32 бита
3	Готовность приемника SPI выдать данные	32 бита
4	Готовность контроллера видеокамеры выдать данные	128 бит
5	Готовность приемника SSI0 контроллера выдать данные	32 бита
6	готовность приемника SSI1 контроллера выдать данные	32 бита
7	Готовность буфера NANDF контроллера выдать данные	32 или 128 (определяется контроллером NANDF)
8	Готовность модуля ADDA0 выдать данные	128 бит
9	Готовность модуля ADDA1 выдать данные	128 бит
10	Готовность модуля ADDA2 выдать данные	128 бит
11	Готовность модуля ADDA3 выдать данные	128 бит
12	-	
13	-	
14	H264_RQ1	128 бит
15	H264_RQ0	128 бит

Каждое из устройств имеет для передачи и (или) для приема встроенный буфер FIFO. При работе с устройствами с помощью каналов 0-3 необходимо задать источник запроса с помощью регистра конфигурации, а также обязательно прописать адрес источника или приемника в соответствующем регистре канала. При работе с каналами 4-11 программируется только номер источника запроса. Особенным образом должны обрабатываться запросы H264\_RQ[1:0]. Модули, формирующие эти запросы, находятся в процессорном ядре и имеют адрес в адресном пространстве внутренней памяти. Поэтому данные запросы могут обрабатываться только с помощью каналов DMA с 0-го по 3-й.

### 10.1.2 Группа регистров TCB каналов 0-3 контроллер DMA

Доступ к регистрам данной группы осуществляется только как к счетверенным словам, т.е. операции выполняются сразу с четырьмя регистрами.

**Таблица 47 – Группа регистров TCB каналов 0-3**

Имя	Описание	Адрес-смещение	Значение по умолчанию
DCS0:DI	канал 0, адрес источника	0x 0000	0
DCS0:DX	канал 0, модификатор X источника	0x 0001	0x01000004
DCS0:DY	канал 0, модификатор Y источника	0x 0002	0
DCS0:DP	канал 0, управление источником	0x 0003	0
DCD0:DI	канал 0, адрес приемника	0x 0004	0
DCD0:DX	канал 0, модификатор X приемника	0x 0005	0x01000001
DCD0:DY	канал 0, модификатор Y приемника	0x 0006	0
DCD0:DP	канал 0, управление приемником	0x 0007	0
DCS1:DI	канал 1, адрес источника	0x 0008	0
DCS1:DX	канал 1, модификатор X источника	0x 0009	0x01000004
DCS1:DY	канал 1, модификатор Y источника	0x 000A	0
DCS1:DP	канал 1, управление источником	0x 000B	0
DCD1:DI	канал 1, адрес приемника	0x 000C	0
DCD1:DX	канал 1, модификатор X приемника	0x 000D	0x01000001

DCD1:DY	канал 1, модификатор У приемника	0x 000E	0
DCD1:DP	канал 1, управление приемником	0x 000F	0
DCS2:DI	канал 2, адрес источника	0x 0010	0
DCS2:DX	канал 2, модификатор Х источника	0x 0011	0x01000004
DCS2:DY	канал 2, модификатор У источника	0x 0012	0
DCS2:DP	канал 2, управление источником	0x 0013	0
DCD2:DI	канал 2, адрес приемника	0x 0014	0
DCD2:DX	канал 2, модификатор Х приемника	0x 0015	0x01000001
DCD2:DY	канал 2, модификатор У приемника	0x 0016	0
DCD2:DP	канал 2, управление приемником	0x 0017	0
DCS3:DI	канал 3, адрес источника	0x 0018	0
DCS3:DX	канал 3, модификатор Х источника	0x 0019	0x01000004
DCS3:DY	канал 3, модификатор У источника	0x 001A	0
DCS3:DP	канал 3, управление источником	0x 001B	0
DCD3:DI	канал 3, адрес приемника	0x 001C	0
DCD3:DX	канал 3, модификатор Х приемника	0x 001D	0x01000001
DCD3:DY	канал 3, модификатор У приемника	0x 001E	0
DCD3:DP	канал 3, управление приемником	0x 001F	0

### 10.1.3 Группа регистров ТСВ каналов 4-7 контроллер DMA

Доступ к регистрам данной группы осуществляется только как к счетверенным словам, т.е. операции выполняются сразу с четырьмя регистрами.

**Таблица 48 – Группа регистров ТСВ каналов 4-7**

<b>Имя</b>	<b>Описание</b>	<b>Адрес-смещение</b>	<b>Значение по умолчанию</b>
DC4:DI	канал 4, адрес источника	0x 0020	Не определено
DC4:DX	канал 4, модификатор Х источника	0x 0021	Не определено
DC4:DY	канал 4, модификатор У источника	0x 0022	Не определено
DC4:DP	канал 4, управление источником	0x 0023	0
DC5:DI	канал 5, адрес источника	0x 0024	Не определено
DC5:DX	канал 5, модификатор Х источника	0x 0025	Не определено
DC5:DY	канал 5, модификатор У источника	0x 0026	Не определено
DC5:DP	канал 5, управление источником	0x 0027	0
DC6:DI	канал 6, адрес источника	0x 0028	Не определено
DC6:DX	канал 6, модификатор Х источника	0x 0029	Не определено
DC6:DY	канал 6, модификатор У источника	0x 002A	Не определено
DC6:DP	канал 6, управление источником	0x 002B	0
DC7:DI	канал 7, адрес источника	0x 002C	Не определено
DC7:DX	канал 7, модификатор Х источника	0x 002D	Не определено
DC7:DY	канал 7, модификатор У источника	0x 002E	Не определено
DC7:DP	канал 7, управление источником	0x 002F	0
-	-	0x0030– 0x003F	-

### 10.1.4 Группа регистров TCB каналов 8-13

Доступ к регистрам данной группы осуществляется только как к счетверенным словам, т.е. операции выполняются сразу с четырьмя регистрами.

К этой группе относятся также и каналы 12 и 13 DMA.

**Таблица 49 – Группа регистров TCB каналов 8-13**

<b>Имя</b>	<b>Описание</b>	<b>Адрес-смещение</b>	<b>Значение по умолчанию</b>
DC8:DI	канал 8, адрес приемника	0x 0040	0x0000 0000
DC8:DX	канал 8, модификатор X приемника	0x 0041	0x0100 0004
DC8:DY	канал 8, модификатор У приемника	0x 0042	0x0000 0000
DC8:DP	канал 8, управление приемником	0x 0043	0x5780 0000
DC9:DI	канал 9, адрес приемника	0x 0044	0x0000 0000
DC9:DX	канал 9, модификатор X приемника	0x 0045	0x0100 0004
DC9:DY	канал 9, модификатор У приемника	0x 0046	0x0000 0000
DC9:DP	канал 9, управление приемником	0x 0047	0x5780 0000
DC10:DI	канал 10, адрес приемника	0x 0048	0x0000 0000
DC10:DX	канал 10, модификатор X приемника	0x 0049	0x0100 0004
DC10:DY	канал 10, модификатор У приемника	0x 004A	0x0000 0000
DC10:DP	канал 10, управление приемником	0x 004B	0x5780 0000
DC11:DI	канал 11, адрес приемника	0x 004C	0x0000 0000
DC11:DX	канал 11, модификатор X приемника	0x 004D	0x0100 0004
DC11:DY	канал 11, модификатор У приемника	0x 004E	0x0000 0000
DC11:DP	канал 11, управление приемником	0x 004F	0x5780 0000
-	-	0x 0050 – 0x 0057	-
DC12:DI	канал 12, адрес приемника	0x 0058	0x0000 0000
DC12:DX	канал 12, модификатор X приемника	0x 0059	0x0100 0001
DC12:DY	канал 12, модификатор У приемника	0x 005A	0x0000 0000
DC12:DP	канал 12, управление приемником	0x 005B	0x5380 0000
DC13:DI	канал 13, адрес приемника	0x 005C	0x0000 0000
DC13:DX	канал 13, модификатор X приемника	0x 005D	0x0100 0001
DC13:DY	канал 13, модификатор У приемника	0x 005E	0x0000 0000
DC13:DP	канал 13, управление приемником	0x 005F	0x5380 0000

Каналы 12 и 13 также называются autoDMA-каналами в связи с тем, что при выполнении записи данных в них ведущим устройством, они автоматически выполняют пересылку принятых данных в запрограммированный в них приемник. Для выполнения операций записи в каналы 12 и 13 используются определенные регистры.

### 10.1.5 Группа регистров AutoDMA

Регистры данной группы относятся к каналам 12 и 13 контроллер DMA и используются для реализации режима «подчиненный» (slave) DMA. Данные регистры могут быть доступны через адресное пространство периферийных устройств.

Таблица 50 – Группа регистров AutoDMA

Имя	Описание	Адрес-смещение	Значение по умолчанию
AUTODMA0:0	Регистр AutoDMA 0, разряды данных 31–0	0x 03E0	Не определено
AUTODMA0:1	Регистр AutoDMA 0, разряды данных 63–32	0x 03E1	Не определено
AUTODMA0:2	Регистр AutoDMA 0, разряды данных 95–64	0x 03E2	Не определено
AUTODMA0:3	Регистр AutoDMA 0, разряды данных 127–96	0x 03E3	Не определено
AUTODMA1:0	Регистр AutoDMA 1, разряды данных 31–0	0x 03E4	Не определено
AUTODMA1:1	Регистр AutoDMA 1, разряды данных 63–32	0x 03E5	Не определено
AUTODMA1:2	Регистр AutoDMA 1, разряды данных 95–64	0x 03E6	Не определено
AUTODMA1:3	Регистр AutoDMA 1, разряды данных 127–96	0x 03E7	Не определено
-	-	0x 03E8– 0x 03FF	-

## 10.2 Настройка DMA-передатч

DMA-операции могут программироваться ядром процессора или внешним хост-устройством. Операция программируется записью в регистры TCB. Регистр TCB – регистр счетверенных слов, описывающий передачу DMA-блока. DMA-канал настраивается записью счетверенного слова в каждый из регистров TCB. Каждый регистр должен быть загружен начальным адресом для блока, приращением адреса, количеством слов и управляющей информацией.

Если TCB запрограммирован на генерацию прерывания, запрос прерывания формируется, когда блок данных полностью передан.

Регистры TCB могут быть доступны только как счетверенные слова.

### 10.2.1 Регистры блока управления передачей (TCB)

Каждый TCB-регистр имеет длину 128 бит и разделен на четыре 32-битных регистра:

- регистр индекса (DI);
- регистр X количества и приращения (DX);
- регистр Y количества и приращения (DY);
- регистр управления и указателя цепочки (DP).

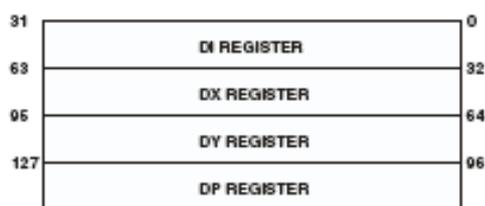


Рисунок 15 – Регистры блока управления передачей

Эти четыре регистра образуют счетверенное слово TCB, которое может быть загружено как выровненное счетверенное слово из внутренней памяти посредством цепочки или с помощью ядра.

Для инициирования нового обмена, после завершения текущего, программа должна записать новые параметры в регистры TCB.

**Таблица 51 – Каналы DMA и связанные регистры TCB**

Канал DMA	Регистры TCB	Описание
0	DCS0	TCB регистр источника канала 0
	DCD0	TCB регистр назначения канала 0
1	DCS1	TCB регистр источника канала 1
	DCD1	TCB регистр назначения канала 1
2	DCS2	TCB регистр источника канала 2
	DCD2	TCB регистр назначения канала 2
3	DCS3	TCB регистр источника канала 3
	DCD3	TCB регистр назначения канала 3
4	DC4	TCB регистр источника канала 4
5	DC5	TCB регистр источника канала 5
6	DC6	TCB регистр источника канала 6
7	DC7	TCB регистр источника канала 7
8	DC8	TCB регистр назначения канала 8
9	DC9	TCB регистр назначения канала 9
10	DC10	TCB регистр назначения канала 10
11	DC11	TCB регистр назначения канала 11
12	DC12	TCB регистр назначения канала 12
13	DC13	TCB регистр назначения канала 13

### **Регистр DI**

32-разрядный регистр DI содержит начальный адрес блока данных и используется для прямого доступа к памяти. Он может указывать на адрес внешней памяти или внутренней памяти.

### **Регистр DX**

32-разрядный регистр DX имеет два поля:

- DXM – младшие биты (с 0-го по 15-й или с 0-го по 21-й) хранят значение модификатора адреса (DX\_MODIFY), которое используется для изменения адреса после каждой транзакции.
- DXC – старшие биты (с 16-го по 31-й или с 22-го по 31-й) хранят значение количества слов (DX\_COUNT) в блоке данных который необходимо передать.

Например, если необходимо передать 16 слов данных, непрерывно размещенных в памяти, порциями по 4 слова в каждой передаче, параметр DXC будет равен 16, а значение DXM будет равно 4. Длина операнда (порции данных) в регистре DP устанавливается равной счетверенному слову.

Существуют ограничения, которые следует принимать во внимание при программировании TCB:

- указатель DI должен содержать адрес, выровненный на границе операнда определяемого в регистре DP;
- значение DXM должно быть кратно размеру операнда;

- значение DXC должно быть кратным длине операнда.

Выбор длины полей модификатора и количества определяется типом обмена, задаваемым полем TY в регистре DP.

#### **Регистр DY**

32-разрядный регистр DY используется вместе с регистром DX, когда двумерный режим DMA разрешен. Регистр DY содержит два поля:

- DYM – младшие биты (с 0-го по 15-й или с 0-го по 21-й) хранят значение модификатора адреса (DY\_MODIFY), которое используется для изменения адреса после завершения передачи блока по координате X. Модификация Y – разница между адресом последнего элемента данных в строке и адресом первого элемента в следующей строке.
- DYC – старшие биты (с 16-го по 31-й или с 22-го по 31-й) хранят значение количества попыток (DY\_COUNT) передач по координате X.

Если двумерный (2D) режим DMA запрещен, регистр DY загружается значением 0 или любым другим допустимым значением.

Выбор длины полей модификатора и количества определяется типом обмена, задаваемым полем TY в регистре DP.

#### **Регистр DP**

Данный регистр задает режим работы передатчика\приемника канала. Подробное описание разрядов регистра приведено ниже (Таблица 52).

**Таблица 52 – Описание разрядов регистра DP**

<b>Бит</b>	<b>Имя</b>	<b>Назначение</b>
18:0	CHPT	Указатель цепочки. Эти разряды включают в себя разряды 20–2 адреса внутренней памяти, где находится значение следующего регистра TCB. Разряды 1–0 адреса не указываются, поскольку адрес должен быть выровненным на границе квадрослова, т.е биты равны нулю. Например, если содержимое следующего TCB сохраняется в памяти 0x87128–0x8712B тогда поле CHPT будет содержать b0010_0001_1100_0100_1010 (0x21C4A).
21:19	CHTG	Канал-приемник следующей цепочки. Это поле имеет значение <u>только для каналов 4-11</u> и указывает в какой канал DMA будет загружено новое значение TCB: 000 – канал 8 001 – канал 9 010 – канал 10 011 – канал 11 100 – канал 4 101 – канал 5 110 – канал 6 111 – канал 7
22	CHEN	Разрешение загрузки следующей цепочки: 1 – разрешено 0 – запрещено При разрешении загрузки после окончания передачи всего блока данных канал выполнит чтение новой TCB из внутренней памяти и загрузит ее в канал назначения.
23	DRQ	Разрешение анализа запроса от периферийного устройства: 1 – разрешено

		0 – запрещено Если разрешено, канал выполняет одну транзакцию только по запросу. Если запрещено, канал выполняет передачу всего блока без ожидания запроса. Бит имеет значение для каналов 0-3 и позволяет им анализировать запросы от устройств. Для каналов 4-11 бит не имеет значения, т.к. они всегда работают только по запросу.
24	INT	Генерация запроса прерывания после окончания работы канала: 1 – разрешено 0 – запрещено
26:25	LEN	Длина передаваемых данных (операнда) в одном цикле обмена: 00 – резерв 01 – слово 32 бита 10 – длинное слово 64 бита 11 – квадрослово 128 бит Длина операнда накладывает ограничения на адрес источника или приемника – они всегда должны быть выравнены. Также накладываются ограничение на модификаторы и количество передаваемых слов в X.
27	2D	Включение режима 2-х мерной пересылки: 1 – двумерная пересылка 0 – одномерная пересылка
28	PR	Приоритет циклов обмена: 1 – высокий 0 – обычный Уровень приоритета используется во время арбитража каналов между собой. Также уровень приоритета оказывает влияние на контакт DPA внешней шины.
31:29	TY	Тип обмена (выбор источника или приемника): 000 – канал выключен 001 – линк порт (только в случае передачи линк-линк) 010 – внутренняя память (16\16) 011 – внутренняя память (22\10) 100 – внешняя память (16\16) 101 – 110 – загрузочное EPROM. nBMS контакт. 111 – внешняя память (22\10)

**Ограничения при программировании регистра DP**

Неправильное задание конфигурации TCB может вызвать генерацию ошибки каналом. На использование *регистра управления TCB* накладываются *следующие ограничения*:

- Установка длины операнда (LEN). В каналах 0-3 поле LEN должно быть одинаковым в обоих TCB.
- Если поле TY источника или получателя TCB имеет тип 6 (загрузочная EPROM), поле LEN должно быть 0x1 (стандартное слово).

Задание полей количества (DX и DY) имеет следующие ограничения:

- Для каналов 0-3 общее количество слов передатчика должно равняться количеству слов приемника.
- Для обычного режима обмена количество слов равно DXC.
- Для двухмерного режима количество слов равно DXC•DYC. При этом бит 2D может быть установлен в одном из TCB и сброшен в другом (для каналов с 0-го по 3-й).

- Нет ограничений для регистра количества в DMA-каналах 4-11. Во всех случаях, если канал не простаивает, DXC не может быть нулевым. Если 2D режим установлен, DYC не может быть 0 или 1

Из поля TY регистра управления видно, что можно задать семь типов обмена. Однако в зависимости от номера канала существует *ограничения в выборе типа*. Разрешенные комбинации (отмечены +) типов в каналах 0-3 приведены ниже (Таблица 53).

**Таблица 53 – Допустимые комбинации передач каналов с 0 по 3**

Тип передатчика	Тип приемника						
	1	2	3	4	5	6	7
1							
2				+		+	+
3				+		+	+
4		+	+	+	+	+	+
5				+			+
6		+	+				
7		+	+	+	+	+	+

Все другие комбинации при их установке в каналах 0-3 вызовут генерацию ошибки канала.

Для каналов 4-7 также существует ограничение в выборе типа обмена: в качестве источника данных могут быть выбраны только внутренняя (тип 2, 3) или внешняя (тип 4, 7) память.

Ограничения в выборе типа для каналов 8-11: в качестве приемника данных могут быть выбраны только внутренняя (2, 3), внешняя (4, 7) память или порт связи (тип 1).

Для канала 12 единственным типом может быть только внутренняя память (тип 2, 3).

**Ограничения запросов DMA**

В каналах 0-3 бит DRQ должен быть одинаковым в обоих регистрах TCB.

**Ограничения выравнивания**

Если поле LEN равно 2 (двойное слово), следующие поля должны быть четными (бит 0 очищен):

- DI;
- DXM;
- DYM (при необходимости);
- DXC.

Если поле LEN равно 3 (счетверенное слово) значения этих же полей должны быть кратны 4 (биты 1–0 очищены).

Для типа 6 (загрузочная EPROM) данные регистры должны быть все кратны 4 (биты 1–0 очищены).

**Ограничения диапазона адресов**

Следующая связь должна существовать в любой момент работы TCB между полем TY в регистре DP и адресом:

- TY = 1 (канал связи) ⇒ адрес – один из регистров передачи порта связи.

- ТУ = 2 или 3 (внутренняя память) ⇒ адрес – в диапазоне внутренней памяти и не в регистрах.
- ТУ = 4 или 7 (внешняя память) ⇒ адрес – в диапазоне внешней памяти (биты 31–22 отличны от нуля).

### **10.2.2 Регистр статуса DMA (DSTAT/DSTATC)**

Регистр статуса DSTAT предназначен только для чтения и отражает текущее состояние каналов контроллера. Регистр имеет разрядность 64 бита и два адреса доступа. Второму адресу соответствует имя DSTATCL регистра. При чтении регистра DSTATCL все коды ошибок в регистре статуса очищаются, и состояние канала изменяется на состояние выключено. Регистры статуса DMA (DSTAT, DSTATCL) могут быть доступны только как двойное или счетверенное слово. В регистре состояния каждому каналу контроллера отведено 3 бита. С помощью этих бит кодируется состояние канала. Возможные значения состояния канала:

- 000 – выключен;
- 001 – активен (работает);
- 010 – завершил работу;
- 011 – резерв;
- 100 – ошибка. Инициализация активного канала;
- 101 – ошибка. Запрещенная конфигурация канала;
- 110 – резерв;
- 111 – ошибка. Некорректный адрес.

Назначение разрядов регистра состояния приведено ниже (Таблица 54).

**Таблица 54 – Биты регистра состояний**

<b>Бит</b>	<b>Имя</b>	<b>Описание</b>
2:0	CH0	Состояние канала 0
5:3	CH1	Состояние канала 1
8:6	CH2	Состояние канала 2
11:9	CH3	Состояние канала 3
14:12	CH4	Состояние канала 4
17:15	CH5	Состояние канала 5
20:18	CH6	Состояние канала 6
23:21	CH7	Состояние канала 7
31:24	-	-
34:32	CH8	Состояние канала 8
37:35	CH9	Состояние канала 9
40:38	CH10	Состояние канала 10
43:41	CH11	Состояние канала 11
49:44	-	-
52:50	CH12	Состояние канала 12
55:53	CH13	Состояние канала 13
63:56	-	-

Активный статус DMA устанавливается, когда канал DMA включен (когда поле ТУ регистра DPx установлено).

Статус завершения DMA устанавливается, когда последняя транзакция DMA завершена.

Статус ошибки DMA устанавливается, если обнаружено запрещенное или ошибочное состояние.

В случае, если обнаруживается ошибочное состояние, DMA может генерировать аппаратное прерывание, если оно включено, и биты статуса могут быть очищены только чтением из регистра DSTATC. Регистры DSTAT и DSTATC должны читаться как сдвоенное или счетверенное слово. Чтение этих регистров как стандартное слово недопустимо.

### 10.2.3 Регистр управления DMA

Биты регистра управления выполняют одну единственную функцию – приостановка работы соответствующего им канала DMA. Подробное описание разрядов регистра управления приведено ниже (Таблица 55).

**Таблица 55 – Биты регистра управления**

Бит	Имя	Описание
0	DMA0_P	Пауза (если 1) для канала 0 DMA
1	DMA1_P	Пауза (если 1) для канала 1 DMA
2	DMA2_P	Пауза (если 1) для канала 2 DMA
3	DMA3_P	Пауза (если 1) для канала 3 DMA
4	DMA4_P	Пауза (если 1) для канала 4 DMA
5	DMA5_P	Пауза (если 1) для канала 5 DMA
6	DMA6_P	Пауза (если 1) для канала 6 DMA
7	DMA7_P	Пауза (если 1) для канала 7 DMA
9:8		
10	DMA8_P	Пауза (если 1) для канала 8 DMA
11	DMA9_P	Пауза (если 1) для канала 9 DMA
12	DMA10_P	Пауза (если 1) для канала 10 DMA
13	DMA11_P	Пауза (если 1) для канала 11 DMA
15:14		
16	DMA12_P	Пауза (если 1) для канала 12 DMA
17	DMA13_P	Пауза (если 1) для канала 13 DMA
31:18		

Регистр управления DMA имеет три адреса:

- DCNT – регулярный адрес для чтения и записи.
- DCNTST – адрес для установки бит. Доступен только по записи. Запись 1 устанавливает соответствующий бит, а запись 0 не изменяет его значения.
- DCNTCL – адрес для очистки бит. Доступен только по записи. Запись 0 сбрасывает значение бита, а запись 1 не изменяет его значение.

Назначение бита паузы состоит в том, что с его установкой в 1 соответствующий канал DMA приостанавливает свою работу и возобновляет ее только после того, как бит станет равен 0.

Регистры управления DMA (DCNT, DCNTST, DCNTCL) могут быть доступны как нормальное, удлиненное и счетверенное слово. Начальное значение DCNT после сброса равно 0.

Управление битом паузы каналами 4-12 довольно простое, т.к. установка бита паузы для них означает остановку генерации транзакций каналом. При этом уже запущенные каналами транзакции спокойно завершаются.

Управление битом паузы для каналов 0-3 более сложное. Это связано с тем, что транзакции этих каналов состоят из двух самостоятельных частей: транзакции чтения данных из источника в буфер DMA и затем транзакции записи данных из буфера по адресу приемника. Канал может запускать до 8-ми транзакций чтения, прежде чем хотя бы одна транзакция записи выполнится. Бит паузы влияет только за запуск транзакций чтения. Но установка бита паузы не означает, что канал остановил работу. Это будет сделано, только когда все запущенные транзакции чтения загрузят в буфер данные, и затем выполнятся все соответствующие им транзакции записи. Изменение настроек канала после установки бита паузы приведет к потере прочитанных данных, т.к. операции записи могут быть сброшены. Чтобы этого не произошло, после установки бита паузы для каналов 0-3 оценивается наличие у канала DMA незавершенных транзакций записи, и если таковые имеются – канал генерирует сигнал блокировки доступа к SOC шине со стороны ядра. Таким образом, если процессор сразу после записи бита паузы пытается произвести операцию на SOC-шине, он может быть остановлен на некоторое время. Данное время может быть значительным, если канал работает с внешней памятью. Это обстоятельство нужно учитывать при работе с каналами 0-3. По возможности не стоит использовать бит паузы для каналов 0-3, если источником данных для каналов является внешняя память и при этом необходима высокая скорость реакции на прерывание.

## **10.3 Операции DMA контроллера**

### **10.3.1 DMA управление каналами 4-11**

Внутренние периферийные устройства процессора способны использовать 13 каналов DMA для обработки приема и передачи данных. Для передачи из внутренней или внешней памяти в устройства используются каналы 0, 2, 4-7. При использовании каналов 4-7 выбор периферийного устройства для требуемого канала выполняется с помощью соответствующих каналу бит регистра конфигурации DMACFGx. Регистры источника TCB программируются адресом внутренней/внешней памяти DI, инкрементом адреса DXM, количеством слов для передачи DXC и управляющими разрядами DP. По окончании программирования TCB автоматически начинается передача и продолжается до завершения передачи блока. При этом транзакции всегда иницируются устройством в момент, когда оно готово принять новую порцию данных. Данный процесс повторяется, пока блок данных не передается полностью.

Для передачи из устройств во внутреннюю или внешнюю память, используются каналы 1, 3, 8-11, 13. При использовании каналов 8-11 выбор периферийного устройства для требуемого канала выполняется с помощью соответствующих каналу бит регистра конфигурации DMACFGx. Регистры приемника TCB программируются адресом внутренней/внешней памяти DI, инкрементом адреса DXM, количеством слов для передачи DXC и управляющими разрядами DP. По окончании программирования TCB автоматически начинается передача. При этом транзакции канала иницируются периферийным устройством в момент, когда у него есть принятые данные. После генерации запроса, принятые данные считываются соответствующим каналом DMA и отсылаются по адресу получателя. Данный процесс повторяется, пока блок данных не передается полностью.

Количество слов для передачи или приема определяется как количество DXC или количество DXC умноженное на количество DYC (если режим 2D установлен).

### 10.3.2 DMA управление каналами 0-3

Особенностью каналов 0-3 является возможность выполнения передачи память-память. В связи с этим каналы 0-3 имеют два регистра ТСВ. Для передачи из внутренней во внешнюю память в регистры источника ТСВ записывается адрес внутренней памяти DI, приращение адреса DXM, количество слов для передачи DXC и управляющие разряды DP. Регистры ТСВ приемника заполняются адресом внешней памяти DI, приращением адреса DXM, количеством слов для передачи DXC и управляющими разрядами DP. При передаче в обратном направлении источник и приемник меняются местами.

После окончания настройки передача начинается автоматически и продолжается до полной передачи блока. Передачи выполняются порциями (по одному операнду на транзакцию). При этом канал может инициировать транзакции непрерывно.

Каналы 0-3 также могут выполнять пересылки устройство-память или память-устройство. В этом случае в ТСВ канала программируются адреса памяти и устройства. Основное отличие в такой передаче это инициирование обмена по запросу устройства. Для этого в регистре конфигурации DMACFGx для заданного канала осуществляется программирование номера устройства. Выбранное устройство будет выполнять инициирование обменов. Для каналов 0-3 возможно задание внешнего источника запроса с помощью выводов nDMAR3-0, т.е. обмен может выполняться с устройством на внешней шине.

Устройство ввода-вывода, в отличие от памяти, сообщает каналу DMA о готовности к передаче данных. Устройство-источник готово, если у него есть данные для записи. Приемник готов, если у него есть место в буфере записи. Для передачи данных между внутренней и внешней памятью могут использоваться данные размером 32, 64 и 128 бит. При доступе к внешней памяти контроллер DMA использует те же интерфейсные сигналы, что и ядро процессора.

### 10.3.3 DMA управление каналом 12

Канал 12 использует режим работы, в котором контроллер DMA выступает в роли ведомого устройства, а цифровой коррелятор (ЦК) – в роли ведущего. Для ЦК канал 12 виден как буфер приема информации с индикацией готовности. Когда ЦК имеет данные для передачи, он анализирует флаг готовности буфера приема канала 12 и записывает в буфер свои данные. Как только в буфере канала появляются данные, канал выполняет их пересылку во внутреннюю память в соответствии с запрограммированным в нем ТСВ. Для передачи данных между ЦК и внутренней памятью могут использоваться внутренние данные размером только 128 бит.

Канал 12 включен, если поле TY в регистре DP не равно нулю. Канал может поддерживать цепочки операций.

Если происходит запись в канал 12, когда он выключен и не занят загрузкой нового ТСВ цепочки, записанные данные теряются, происходит прерывание аппаратной ошибки и ошибка отражается в регистре SYSTAT.

Если запись в канал происходит, когда он выключен, но занят загрузкой ТСВ новой цепочки, данные помещаются в буфер и передаются по назначению, когда загрузка ТСВ для канала завершается. В этом случае нет генерации ошибки. В ТСВ канала программируется только базовый адрес буфера приемника. Смещение относительно базы передает сам ЦК вместе с соответствующей ему порцией данных. Канал 12 формирует итоговый адрес, суммируя значения базы и смещения.

Канал 12 работает с ЦК, только если установлен бит DRQ в регистре управления TCB. Если данный бит сброшен, канал 12 может выполнять обмен в случае, когда в его регистр autoDMA производит запись ведущее устройство.

## **10.4 DMA передача**

### **10.4.1 Адресация памяти**

Контроллер DMA поддерживает словную адресацию памяти. Это означает, что минимальной адресуемой единицей является 32-разрядное слово, и адрес в 32-разрядном индексном регистре есть адрес слова.

Благодаря 32-разрядному регистру адреса DMA может получать доступ к полному адресному пространству процессора. Каждый канал включает регистр индекса DI и поле модификации DXM в своем регистре TCB, которые используются для адресации буфера данных в памяти. 32-битный регистр TCB DI содержит начальный адрес блока и должен инициализироваться стартовым адресом буфера данных. Поле модификации DXM содержит значение со знаком (т.е. положительное или отрицательное) размером 16 или 22 бита и определяет величину изменения адреса после выполнения каждой транзакции. Измененное значение адреса используется в следующей транзакции. Само значение адреса в регистре DI не указывает на адресацию внутренней или внешней памяти. Чтобы указать, какому типу памяти соответствует адрес, используется дополнительное поле TY в регистре управления DP блока TCB. Если в результате модификации, адрес выйдет за границы существующей памяти или сменит область внутренней памяти на внешнюю (или наоборот), контроллер DMA все равно будет обращаться к области памяти в соответствии с полем TY.

Передаваемые данные могут быть стандартными, сдвоенными или счетверенными словами. Это указывается в поле LEN регистра управления DP.

Каждый канал DMA имеет поле количества DXC в регистре DX, которое должно быть инициализировано количеством слов для передачи, независимо от длины данных (32, 64 или 128 бит). Счетчик количества уменьшается после каждой передачи DMA в данном канале. Значения уменьшения равны 1, 2 или 4 и определяются полем LEN. Когда количество достигает значения ноль, DMA завершается и может быть сгенерировано прерывания для информирования процессора.

Если поле количества DXC в TCB DX инициализируется нулем, DMA передача на этом канале возможна и будет соответствовать выполнению передачи максимально возможного количества данных. Это происходит из-за того, что первая передача начинается до того как проверяется значение количества.

Значение количества анализируется после вычитания из него 1. Надежным способом отключения DMA-канала является сброс поля TY в соответствующем управляющем регистре.

### **10.4.2 Приоритеты каналов DMA**

Система приоритетов используется при выборе канала DMA для обслуживания, т.к. в одном такте может поступить активный запрос более чем от одного канала DMA. В зависимости от бита PR в регистре DP блока TCB канал может относиться к группе каналов с высоким приоритетом или к группе с низким приоритетом. Внутри каждой группы каналов используется фиксированная система приоритетов между каналами. Описание каналов DMA в порядке уменьшения приоритетов приведено ниже (Таблица 56).

Приоритет устанавливается исходя из следующих соображений:

- Входящие данные имеют более высокий приоритет, чем исходящие. Приемники устройств имеют более высокий приоритет, чем передатчики.
- Каналы устройств имеют более высокий приоритет, чем каналы общего назначения.
- Каналы общего назначения дополнительно имеют циклически изменяемый приоритет в последовательности 3-2-1-0.

Контроллер DMA анализирует запросы от всех каналов внутри двух групп и в течение одного такта определяет канал с самым высоким приоритетом. Выбранный канал внутри высокоприоритетной группы имеет преимущество над каналом обычной группы. Выбранному каналу в текущем такте предоставляется право выполнить транзакцию. В следующем такте арбитраж повторяется.

Загрузка цепочки TCB получает приоритет канала. Например, передача данных канала с более высоким приоритетом производится перед загрузкой цепочки TCB с более низким приоритетом.

**Таблица 56 – Приоритеты каналов DMA**

Канал	Описание	Приоритет
Канал 13	Хост-интерфейс Интерфейс ЦК	Наиболее высокий
Канал 12		
Канал 11	Устройства-приемники	
Канал 10		
Канал 9		
Канал 8		
Канал 7	Устройства-передатчики	
Канал 6		
Канал 5		
Канал 4		
Канал 3	Каналы общего назначения	
Канал 2		
Канал 1		
Канал 0		Наиболее низкий

#### **10.4.3 Циклически присваиваемый приоритет**

Для каналов 0-3 сделана дополнительная модификация – циклический приоритет внутри их группы. Это означает, что после каждого цикла арбитража приоритет каналов 0-3 меняется циклически. Приоритеты каналов после сброса следующие:

- канал 3 (самый высокий);
- канал 2;
- канал 1;
- канал 0 (самый низкий).

Каждый раз, если одному из каналов 0-3 предоставляется возможность выполнить транзакцию, приоритет каналов меняется. Только что работавший канал получает высочайший приоритет, остальные приоритеты расставляются как описано выше, но по циклу. Например, если канал 1 в текущем такте выполняет транзакцию, в следующем такте приоритет каналов следующий:

- канал 1 (самый высокий);
- канал 0;
- канал 3;
- канал 2 (самый низкий).

Если следующую транзакцию выполняет канал 0, приоритеты следующие:

- канал 0 (самый высокий);
- канал 3;
- канал 2;
- канал 1 (самый низкий).

Приоритеты каналов 0-3 остаются неизменными при выборе каналов с более высоким приоритетом (4-13). По окончании транзакции канала с более высоким приоритетом, порядок приоритетов в каналах 0-3 сохраняется прежним.

В примерах выше подразумевалось, что все четыре канала не устанавливают бит приоритета в ТСВ или все устанавливают, т.е. принадлежат одной группе. Если один или более канал установлен на высокий приоритет, циклический алгоритм применяется для каждой группы в отдельности. Например, если после сброса у канала 2 установлен бит приоритета, приоритеты следующие:

- канал 2 (установлен бит);
- канал 3 (самый высокий);
- канал 1;
- канал 0 (самый низкий).

Если канал 1 запрашивает доступ и получает его, схема приоритетов следующая:

- канал 2 (установлен бит);
- канал 1 (самый высокий);
- канал 0;
- канал 3 (самый низкий).

Если в течение работы канала 1, канал 2 генерирует запрос, он получает доступ и работа канала 1 прерывается. После окончания передачи каналом 2 схема приоритетов для каналов с более низкими приоритетами не меняется, и канал 1 выигрывает арбитраж и продолжает передачу. Если более одного канала имеют установленный бит приоритета, арбитраж между этими каналами проходит циклически.

#### **10.4.4 Приоритет контроллер DMA на внутренних шинах процессора**

Канал DMA, выигравший арбитраж приоритетов среди всех каналов внутри контроллера, получает возможность выполнить транзакцию и генерирует запрос к одному из трех ресурсов системы (Рисунок 12):

- SOC IFIFO – входной буфер SOC-интерфейса для доступа к внутренней памяти.
- EBIU OFIFO – выходной буфер внешнего интерфейса для доступа к внешней памяти.
- SOC-шина периферийных устройств.

При арбитраже запросов к SOC IFIFO контроллер DMA имеет самый низкий приоритет. Первыми получают доступ данные, считываемые процессором с SOC-шины, или запрос от EBIU IFIFO внешнего интерфейса.

При арбитраже запросов к EBIU OFIFO контроллер DMA также имеет самый низкий приоритет. Первым получает доступ ядро процессора. При этом не имеет значения высокоприоритетный запрос DMA или нет.

При арбитраже запросов к SOC-шине периферийных устройств DMA имеет более высокий приоритет, чем ядро процессора, но ниже чем возвращаемые из ядра или внешней памяти данные. Данные, возвращаемые из ядра или внешней памяти, являются результатом пересылки DMA типа память-устройство. Канал DMA формирует запрос к SOC-шине только при выполнении пересылки устройство-память.

#### **10.4.5 Цепочка DMA**

Традиционный алгоритм работы DMA заключается в программировании процессором канала DMA, выполнение каналом передачи данных и генерации прерывания после завершения работы. В ответ на запрос прерывания процессор может запрограммировать канал на новую передачу.

Использование цепочек в DMA позволяет контроллеру автоматически инициализировать себя следующей передачей блока данных после завершения передачи текущего блока. При этом участие процессора не требуется, и он экономит время, которое потратил бы на обработку прерывания и программирование нового обмена. Используя механизм цепочки, могут быть организованы множественные операции DMA с различными атрибутами и устройствами ввода-вывода.

В первую очередь для поддержки цепочки используются биты регистра TCB DP:

- Бит CHEN разрешения цепочки.
- Поле СНРТ указателя на цепочку. Это адрес в памяти, который содержит следующий TCB для загрузки (где содержится адрес, инкремент, количество слов для передачи и управляющие разряды для следующего передаваемого блока).
- Поле СНТГ приемника нового TCB. Прочитанное из памяти значение TCB загружается в канал DMA указанный в данном поле.

Таким образом, в каждом TCB канала имеется механизм для загрузки в регистр TCB нового значения командного слова после окончания работы. Из-за ограничения размера указателя хранение значений цепочек возможно только во внутренней памяти процессора.

Существует несколько ограничений на цепочки DMA между каналами:

- Для каналов 0-3 возможна поддержка только одноканальной цепочки, т.е. поле СНТГ не имеет значения, и новый TCB цепочки может быть загружен только в вызвавший ее канал. При этом в обоих TCB канала бит CHEN должен быть установлен, т.к. для работы этих каналов нужна загрузка двух TCB.
- Для каналов DMA 4-11 возможна межканальная цепочка, когда один канал может запускать работу в другом канале. Например, канал 8, приняв блок данных и загрузив его в память, может инициировать канал 4 на передачу принятых данных.

### **Включение и выключение цепочек**

Передачи DMA запускаются записью счетверенного слова в регистр TCB (каналы 0-3 требуют загрузки обоих регистров TCB). Цепочка образуется при установленном бите разрешения CHEN, и если указатель цепочки СНРТ является разрешенным адресом.

Поле СНРТ в TCB DP может быть загружено:

- при инициализации канала;
- во время работы канала (вставка цепочки).

В первом варианте пользователь заранее знает, что необходимо выполнить несколько разнообразных передач, и программирует цепочку операций, загружая первый TCB в канал, а все последующие TCB сохраняя во внутренней памяти.

Во втором случае пользователь запускает передачу одного блока, но в процессе выполнения программы (возможно, другим процессом) возникает необходимость в передаче нового блока данных. В этом случае программа определяет, что требуемый канал активен и устанавливает для данного канала бит паузы. Это вызывает приостановку работы канала и позволяет программе:

- Если цепочка операций в активном TCB не включена – организовать вставку цепочки путем установки бита CHEN в 1 и загрузки поля указателя СНРТ необходимым значением. Значение TCB новой передачи сохраняется во внутренней памяти.
- Если цепочка операций включена – добавить в конец (или в любое место) списка цепочек свою запись.

После выполнения описанных действий процессор может сбросить бит паузы и канал продолжит работу. После сброса бита паузы приостановленный активный канал выполняет повторный контроль своего состояния и, в случае корректной работы, продолжает работу.

### **Генерация прерывания в цепочке операций**

Пользователь может сам выбрать режим прерывания при работе цепочки. Если необходима генерация прерывания при передаче конкретного блока, бит INT в регистре TCB блока нужно установить в 1. Прерывание произойдет после передачи данного блока. Если необходимо сгенерировать прерывание только после окончания работы всей цепочки, бит INT должен быть установлен только в последнем TCB цепочки.

### **TCB и загрузка цепочек**

Во время загрузки TCB цепочки, регистры TCB канала DMA загружаются значениями, полученными из внутренней памяти. TCB хранится в четырех последовательных словах выровненного счетверенного слова. Чтение нового TCB цепочки иницируется после передачи всего блока данных. Для каналов 0-3 необходимо чтение 2-х TCB. При необходимости чтения цепочки канал формирует запрос на чтение TCB со своим приоритетом, т.е. чтение цепочки каналом имеет такой же приоритет, как и чтение данных.

## **10.4.6 Двухмерный DMA**

Данный раздел описывает изменения в работе, которые происходят при включении режима двухмерного DMA в процессоре. Режим двухмерного DMA включается установкой бита 2D в TCB DP регистре. Этот режим использует внешние и внутренние адреса. Преимущество двухмерной адресации состоит в

перераспределении данных из одномерной строки (вектора) в представление в виде двумерного массива (матрицы) с координатами X и Y. Измерения X и Y определяются в регистрах TCB передатчика и/или приемника. При этом измерение X определяет количество слов в строке массива, а измерение Y – количество строк.

- Измерения массива передатчика и приемника могут отличаться. Главное требование – итоговое число слов в источнике и получателе должно быть равно.
- Двухмерный массив памяти может быть перемещен в одномерный массив и наоборот, если итоговое число слов в источнике и получателе равно.
- Двухмерный блок в памяти может быть передан в порты связи, а блок, полученный из порта связи или из AutoDMA, может быть размещен в памяти как двумерный массив.

#### 10.4.7 Организация канала двумерного DMA

В режиме двумерного DMA адресация к массиву может быть выполнена на любом DMA-канале, приемнике или передатчике. Регистр индекса (DI) загружается адресом первого элемента массива и обновляется после каждой передачи операнда на значение модификатора DXM, пока поле количества DXC не достигнет нуля. Величина модификатора DXM в регистре DX содержит смещение, добавляемое к текущему значению адреса для перехода на следующий элемент в измерении X (следующая колонка).

Для режима 2D поле количества DXC в регистре DX имеет дополнительную буферизацию (внутренний буфер CIX). При загрузке TCB, поле количества DXC и буфер CIX загружаются одинаковым значением. В процессе передачи по измерению X количество в DXC уменьшается до нуля и буфер CIX используется для перезагрузки количества в DXC. Можно сказать, что измерение X работает в режиме 2D так же, как в обычном режиме, только после передачи всей строки X происходит перезагрузка значения количества.

Регистр DY в TCB используется для задания количества строк массива и для перехода от одной строки к другой. В процессе передачи текущей строки к адресу элемента прибавляется значение DXM и осуществляется переход к следующему элементу строки (элементы не обязательно последовательны). При передаче последнего элемента строки (поле DXC достигает значения ноль) к адресу последнего элемента прибавляется модификатор DYM (модификатор DXM не добавляется). Это позволяет перейти к первому элементу следующей строки. Значение модификации DYM в регистре DY является целочисленным числом со знаком, что позволяет уменьшать или увеличивать адрес.

Значение количества строк в DYC уменьшается на 1 и, если это была не последняя строка, количество DXC перезагружается из CIX.

Когда поле количества DYC достигает нуля, DMA-передача блока окончена, образуя матрицу размером X на Y.

#### Пример задания двумерной передачи

Матрица с количеством колонок COL и количеством строк ROW. Каждый элемент матрицы имеет размер TXS (он определяется полем LEN в DP и может быть равен 1, 2 или 4). Элементы матрицы размещаются в памяти последовательно, причем сначала элементы первого столбца, затем второго и т.д. Например, необходимо выполнить передачу массива построчно. Следующая формула может быть применена для расчета параметров DX и DY.

$DY = TXS * COL \ll 16$ ; // количество слов данных в строке  
 $DY |= TXS * ROW$ ; // шаг от текущего элемента строки к следующему  
 $DY = ROW \ll 16$ ; // количество строк  
 $DY |= (-TXS * (((COL - 1) * ROW) - 1)) \& 0xFFFF$ ; // переход к следующей строке

Операция «И» с 0xFFFF использовалась для того чтобы ограничить длину отрицательного модификатора размеров в 16 бит перед операцией логического ИЛИ с количеством строк.

#### Двухмерные DMA операции

Рассмотрим подробно действия, которые происходят внутри канала при двухмерной передаче. Все действия происходят в течение одного такта, но имеют причинно-следственную связь:

- Берется текущий адрес в регистре DI и запускается транзакция передачи операнда длиной LEN (чтение или запись).
- При подтверждении отправки транзакции происходит вычитание из DXC числа слов в операнде. Результат вычитания анализируется.
- Если результат вычитания не равен нулю, к содержимому регистра DI прибавляется значение модификации DXM. Результат вычитания помещается в поле количества DXC, а новое значение адреса в DI. На этом действия канала в текущей транзакции закончены.
- Если результат вычитания равен нулю, происходит вычитание единицы из поля количества DYC. Результат вычитания анализируется.
- Если результат вычитания количества DYC равен нулю – канал завершает работу.
- Если результат вычитания не равен нулю, в регистр количества DXC загружается начальное значение из буфера CIX, к значению регистра DI прибавляется значение модификатора DYM, результат вычитания загружается в поле количества DYC. Процесс повторяется.

Особенность двухмерного DMA (и любого DMA) в том, что первая передача начинается до анализа количества. Это означает, что DMA не может быть выключен установкой в ноль количества в регистре DXC или DYC. Для отключения двухмерного DMA необходимо очистить бит 2D в регистре управления DP. Для отключения канала DMA, необходимо очистить поле TY в регистре DP.

#### 10.4.8 Прерывания DMA

Прерывание окончания DMA генерируется, если бит INT в регистре канала DP установлен.

DMA генерирует прерывание, когда количество передач в активном канале DMA доходит до нуля и передача заканчивается. Когда назначением DMA является внутренняя память, окончание передачи означает, что последний элемент данных попал в буфер SOC IFIFO.

Когда назначением DMA является внешняя память, окончание передачи означает, что последний элемент данных попал в буфер EBIU OFIFO.

Если назначением передачи является устройство-передатчик, окончание передачи означает, что последний элемент данных был загружен в буфер передатчика.

Видно, что генерация прерывания для случая, когда пункт назначения данных в памяти, не означает, что последние данные уже находятся в памяти. Это произойдет с задержкой. Для случая внутренней памяти задержка может достигать:

- 32-х тактов процессора (если запись идет в блок памяти, куда одновременно обращаются с чтением все три внутренних процессорных шины);
- 16-ти тактов (если запись идет в блок памяти без конфликтов).

Минимальная задержка 3 такта. Процессор не в состоянии за это время обработать запрос прерывания и попытаться прочитать последнее слово переданных данных.

Задержка для внешней памяти зависит от частоты и настроек системной шины.

Каждый канал DMA имеет собственный бит запроса прерывания в контроллере прерываний. Прерывания DMA фиксируются в регистре ILAT и разрешаются в регистре IMASK. Приоритет запросов прерываний от каналов DMA фиксированный и определен в контроллере прерываний.

Опрос регистра DSTAT может использоваться как альтернатива прерываниям для определения окончания одиночной последовательности DMA. Если включены цепочки, опрос DSTAT нельзя использовать, т.к. следующая последовательность DMA может быть на подходе в момент возврата статуса по результатам опроса.

#### **10.4.9 Запуск и окончание последовательностей DMA**

Данный подраздел рассматривает запуск, приостановку, возобновление и окончание последовательностей DMA.

##### **Запуск последовательности DMA**

Последовательность DMA для каналов 0-3 запускается в одном из случаев:

- включен канал DMA, бит запроса DRQ равен нулю;
- включен канал x DMA, установлен бит запроса DRQ и получен запрос от выбранного устройства.

Последовательность DMA для каналов 4-7 запускается в случае, когда:

- включен канал DMA и имеется запрос от устройства-передатчика на прием данных в буфер передатчика. Бит запроса DRQ не имеет значения.

Последовательность DMA для каналов 8-11 запускается в случае, когда:

- включен канал DMA и имеется запрос от устройства-приемника о наличии данных в буфере приемника. Бит запроса DRQ не имеет значения.

Последовательность DMA для канала 12 запускается в случае, когда:

- Включен канал DMA и имеются данные в буфере канала. Буфер канала может принять до 4-х операндов. Бит запроса DRQ не имеет значения.

Для запуска новой последовательности DMA после завершения текущей, программа должна записать новые параметры в регистры TCB. Запись активного TCB в активный TCB регистр вызывает ошибку.

##### **Приостановка последовательности DMA**

Последовательность операций DMA в некотором канале приостанавливается, когда соответствующий ему бит паузы в регистре DCNT установлен. Установка бита паузы может произойти в любой момент работы канала: канал может работать, загружать цепочку или завершить работу. Поэтому после установки бита паузы пользователь должен перед продолжением работы проанализировать состояние канала.

Если момент установки паузы пришелся на загрузку цепочки, бит паузы не остановит загрузку цепочки, и до момента загрузки цепочки поле ТУ регистра ТСВ будет читаться как 0. Хотя биты состояния в этот момент будут показывать, что канал активен.

Если момент установки бита паузы пришелся на завершение работы каналом, поле ТУ регистра ТСВ будет равно нулю, и регистр состояния укажет, что канал не активен.

Установка бита паузы в момент нормальной работы канала (пересылка данных) вызывает останов операций передачи. При этом каналы с 4 по 12 немедленно прекращают операции пересылки. Каналы с 0 по 3 немедленно прекращают операции чтения данных из источника, но продолжают выполнять незавершенные операции записи.

**Внимание!** При наличии незавершенных операций записи после установки бита паузы SOC-шина становится недоступной для ядра процессора.

Разблокировка доступа произойдет только после завершения операций записи.

Установка бита паузы позволяет записать новое активное значение ТСВ (поле ТУ не равно нулю) в активный регистр ТСВ. Такая запись приводит к генерации аппаратной ошибки. Однако, в случае паузы такая запись разрешена. Если случай записи без бита паузы рассматривается как сбой в программе, запись с установленным битом паузы является осознанной последовательностью действий по смене режима работы канала. Это позволяет отложить задачу выполняемую каналом и запрограммировать канала на более приоритетную задачу. После ее завершения состояние канала может быть восстановлено.

Имеется особенность в перепрограммировании каналов с 0 по 3. Если новое активное ТСВ имеет поле ТУ такое же, как активный регистр ТСВ канала, данная запись рассматривается только как попытка модификации регистра DP (например, попытка вставить цепочку). В этом случае при записи обновляется только значение регистра DP. Все другие значения неизменны. Поэтому для каналов с 0 по 3, для задания нового обмена необходимо:

- сохранить текущие параметры регистра ТСВ;
- записать в регистр ТСВ значение с полем ТУ равным нулю;
- записать активное значение в ТСВ регистр.

#### **Возобновление последовательности DMA**

Последовательность операций DMA в некотором канале возобновляется, когда соответствующий ему бит паузы в регистре DCNT сбрасывается в ноль. Для каналов с 4 по 13 это означает немедленное возобновление работы в соответствии с командой ТСВ. Для каналов с 0 по 3 сброс бита паузы может привести к процедуре проверки правильности конфигурации ТСВ, если во время паузы были записи в ТСВ регистры.

#### **Окончание последовательности DMA**

Последовательность DMA заканчивается в одном из случаев:

- Регистры количества равны нулю и цепочка заканчивается.
- Канал отключен сбросом поля ТУ в одном из ТСВ и DP регистрах. При этом такая запись неактивного ТСВ может происходить в любой момент. Для каналов с 0 по 3 могут быть запущены до 8-ми транзакций чтения во внутреннюю или внешнюю память, и после отключения канала все эти данные поступят во внутренний буфер контроллера DMA и будут утеряны, если канал выключен. Если к этому времени канал будет включен для

новой задачи, оставшиеся от предыдущей задачи данные вызовут ошибку в работе канала, т.к. обнаружится запись данных, которые не читались.

## 10.5 Каналы DMA общего назначения

Четыре канала DMA с 0 по 3 являются каналами общего назначения, т.к. обладают двумя регистрами TCB и позволяют программировать произвольный источник и приемник данных. Регистры TCB каналов определяют следующее:

- канал включен, если оба поля TY в TCB DP не равны нулю;
- приоритет канала высокий, если бит PR установлен в одном из регистров TCB DP;
- прерывание DMA включено, если бит INT установлен в одном из регистров TCB DP;
- режим запроса DMA включен, если бит DRQ установлен в одном из регистров TCB DP;
- поле LEN должно быть одинаковым в обоих регистрах;
- бит CHEN должен быть одинаковым в обоих регистрах;
- передача внутренней и внешней памяти устанавливается в поле TY регистров TCB DP, что позволяет эффективно передавать данные между внутренней памятью процессора и внешней памятью или устройством;
- поле CHTG не имеет смысла для этих каналов. В случае цепочки каждый регистр TCB загружается новое значение в свой канал в соответствии с его полем CHPT.

Передача DMA между внутренней памятью процессора и внешней памятью требует от одного канала DMA генерации адресов для обоих типов памяти. Внешний адрес генерируется в соответствии с данными в регистре TCB внешней памяти. Адрес внутренней памяти генерируется в соответствии с данными в регистре TCB внутренней памяти. Если бит 2D в TCB DP установлен в одном из регистров TCB, адрес этого TCB также обновляется в соответствии со значением модификатора в регистре DY. Передачи внутри внешней памяти и внутренней памяти поддерживаются. Также поддерживается передача между внешней памятью и внешними устройствами.

Поле TY в регистре DP указывает тип передачи. Ранее приводилось описание допустимых комбинаций типов в одном канале (Таблица 46). Каждый канал с 0 по 3 имеет регистр TCB передатчика и приемника, где передатчик считывает данные из памяти и передает их приемнику, а приемник отсылает данные в память.

Большинство параметров TCB не имеют отношения к типу памяти, а являются описателями блока памяти (который может размещаться в любом месте) или описывают поведение канала во время работы и при завершении обмена.

Описатели блока памяти:

- регистры DX и DY задают размеры блока и расположение элементов блока в памяти;
- бит 2D определяет, рассматривается блок данных как линейный или как двухмерный;
- биты LEN определяют длину операнда блока.

Описатели поведения канала во время работы и при завершении обмена:

- бит PR указывает, что канал имеет высокий приоритет;
- бит DRQ определяет активацию транзакции по запросу или без него;

- бит INT разрешает формирование запроса прерывания к процессору по окончании обмена;
- бит CHEN разрешает цепочку операций;
- поле СНРТ хранит адрес цепочки;
- поле СНТГ определяет приемник цепочки.

Рассмотренные выше параметры не зависят от типа обмена. Поэтому далее при описании различных вариантов обмена будут рассматриваться параметры, которые зависят от выбранного типа памяти.

### **10.5.1 Передача из внешней памяти во внутреннюю память**

Для передачи данных из внешней памяти во внутреннюю необходимо запрограммировать TCB передатчика для чтения внешней памяти, а TCB приемника – для записи во внутреннюю память. TCB передатчика программируется следующим образом:

- Регистр DI  
Инициализируется начальным адресом блока данных во внешней памяти.

- Регистр DP  
Поле TY должно быть равно 4 или 7, что указывает на внешнюю память. Выбор между типом 4 и 7 влияет только на длину полей DXC, DXM, DYC, DYM.

TCB приемника данных описывается аналогичным образом со следующими особенностями, связанными с выбором внутренней памяти:

- Регистр DP  
Поле TY должно быть равно 2 или 3, что указывает на внутреннюю память. Выбор между типом 2 и 3 влияет только на длину полей DXC,DXM,DYC,DYM приемника.

- Регистр DI  
Инициализируется начальным адресом блока данных во внутренней памяти.

Все другие параметры TCB для передатчика и приемника описывают блоки памяти и поведение канала. В описанных выше параметрах имеются ограничения, основные из которых:

- поле LEN должно быть одинаковым в обоих TCB;
- бит CHEN должен быть одинаковым в обоих TCB.

Выбор между типами обменов 2 и 3 или 4 и 7 зависит от размера величины модификации адреса. Если при переходе от одного элемента к другому необходимо изменить адрес на величину более чем 32768 (в положительную сторону или отрицательную), недостаточно 16 бит поля DXM для хранения такого значения. Поэтому следует использовать тип обмена 3 или 7 вместо 2 или 4, чтобы поле модификации было 22 бита. В этом случае уменьшается размер поля количества.

### **10.5.2 Передача из внутренней памяти во внешнюю память**

В случае передачи из внутренней памяти во внешнюю память нет никаких особенностей. По сравнению с режимом, описанным в подразделе «Передача из

внешней памяти во внутреннюю память», ТСВ передатчика и приемника просто меняются местами.

ТСВ передатчика:

– Регистр DP

Поле TY должно быть равно 2 или 3, что указывает на внутреннюю память. Выбор между типом 2 и 3 влияет только на длину полей DXC,DXM,DYC,DYM приемника.

– Регистр DI

Инициализируется начальным адресом блока данных во внутренней памяти.

ТСВ приемника:

– Регистр DP

Поле TY должно быть равно 4 или 7, что указывает на внешнюю память. Выбор между типом 4 и 7 влияет только на длину полей DXC,DXM,DYC,DYM приемника.

– Регистр DI

Инициализируется начальным адресом блока данных во внешней памяти.

### **10.5.3 Передача из внешней памяти во внешнюю память**

Подобный тип передачи выполняется в случае, когда в ТСВ приемника и передатчика поля типа TY равны 4 или 7. Адреса в регистрах DI должны указывать на блоки данных во внешней памяти.

### **10.5.4 Передача из внутренней памяти во внутреннюю память**

Данный тип передачи программируется следующим образом.

ТСВ передатчика:

– Регистр DP

Поле TY должно быть равно 2 или 3, что указывает на внутреннюю память. Выбор между типом 2 и 3 влияет только на длину полей DXC,DXM,DYC,DYM приемника.

– Регистр DI

Инициализируется начальным адресом блока данных во внутренней памяти.

ТСВ приемника:

– Регистр DP

Поле TY должно быть равно 2 или 3, что указывает на внутреннюю память. Выбор между типом 2 и 3 влияет только на длину полей DXC,DXM,DYC,DYM приемника.

– Регистр DI

Инициализируется начальным адресом блока данных внутренней памяти.

Необходимо отметить, что пересылки во внутренней памяти с помощью DMA выполняются как минимум вдвое медленнее, чем такие же пересылки выполненные

ядром процессора. Однако полезным может быть тот факт что пересылка выполняется в фоновом режиме и процессор может выполнять в это время другую работу.

#### **10.5.5 Передача из устройства ввода во внутреннюю или внешнюю память**

Данный тип передачи программируется следующим образом.

ТСВ передатчика:

– Регистр DP

Поле TY должно быть равно 4 или 7, что указывает на внешнюю память. Выбор между типом 4 и 7 влияет только на длину полей DXC,DXM,DYC,DYM приемника.

Бит DRQ должен быть установлен, что указывает на инициирование обмена только по запросу от устройства.

– Регистр DI

Инициализируется адресом буфера периферийного устройства. При этом если бит 31-й адреса равен 1, это адрес внутреннего периферийного устройства на SOC-шине. Если бит равен нулю, это внешнее устройство.

ТСВ приемника:

– Регистр DP

Поле TY должно быть равно 2 (3) или 4 (7), что указывает на внутреннюю или внешнюю память. Выбор влияет только на длину полей DXC,DXM,DYC,DYM приемника.

Бит DRQ должен быть установлен, что указывает на инициирование обмена только по запросу от устройства.

– Регистр DI

Инициализируется начальным адресом блока данных внутренней или внешней памяти.

Регистр конфигурации DMACFGx задает номер периферийного устройства, которое будет формировать сигналы запроса на обмен.

#### **10.5.6 Передача из внутренней или внешней памяти в устройство вывода**

Данный тип передачи программируется следующим образом.

ТСВ передатчика:

– Регистр DP

Поле TY должно быть равно 2 (3) или 4 (7), что указывает на внутреннюю или внешнюю память. Выбор типа влияет только на длину полей DXC,DXM,DYC,DYM приемника.

Бит DRQ должен быть установлен, что указывает на инициирование обмена только по запросу от устройства.

– Регистр DI

Инициализируется начальным адресом блока данных во внутренней или внешней памяти.

ТСВ приемника:

– Регистр DP

Поле TY должно быть равно 4 или 7, что указывает на внешнюю память. Выбор между типом 4 и 7 влияет только на длину полей DXC,DXM,DYC,DYM приемника.

Бит DRQ должен быть установлен, что указывает на инициирование обмена только по запросу от устройства.

– Регистр DI

Инициализируется адресом буфера периферийного устройства. При этом если бит 31-й адреса равен 1, это адрес внутреннего периферийного устройства на SOC-шине. Если бит равен нулю, это внешнее устройство.

Регистр конфигурации DMACFGx задает номер периферийного устройства, которое будет формировать сигналы запроса на обмен.

Канал DMA не контролирует корректность соответствия номера устройства в регистре DMACFGx адресу буфера устройства в регистре ТСВ. Периферийное устройство выступает только в роли инициатора транзакции. Однако пересылка данных определяется только ТСВ источника и приемника. Поэтому по инициативе периферийного устройства с помощью каналов 0-3 можно выполнять любой вариант пересылки, который можно запрограммировать в канале. Отличие каналов 0 и 2 от каналов 1 и 3 только в том, что они имеют различные источники запросов обмена. При этом нужно помнить, что если какое-то устройство сформировало запрос на обмен, для формирования следующего запроса устройство должно быть кем-то обслужено, т.е. из источника должны быть прочитаны данные, а в приемник записаны.

## 10.6 Каналы DMA с 4 по 11 для работы с устройствами

Четыре канала DMA определяют передачи от устройств к внутренней памяти или внешней памяти. Другие четыре канала определяют передачи из внутренней памяти или внешней памяти в устройства.

Особенностью каналов DMA является то, что все они имеют только один регистр ТСВ. Для каналов с 4 по 7 это ТСВ передатчика, который читает данные из внешней или внутренней памяти и отправляет данные в устройство-передатчик. Для каналов с 8 по 11 это ТСВ приемника, который берет данные из устройства-приемника и пересылает их во внутреннюю или внешнюю память. Наличие только одного ТСВ регистра в каналах 4-11 не позволяет определить вторую сторону пересылки (источник или приемник). Для разрешения данной проблемы и используется дополнительный регистр DMACFGx.

Как в случае каналов общего назначения, большинство параметров ТСВ не имеют отношения к устройству и описывают только блок данных или поведение канала во время работы или при завершении передачи. Поле TY регистра ТСВ указывает на тип передачи.

### 10.6.1 Передача из устройства во внутреннюю \ внешнюю память

Для переноса данных от устройства во внутреннюю или внешнюю память должны использоваться только каналы с 8 по 11. Приемник ТСВ этих каналов должен быть запрограммирован соответствующим образом.

Для передачи во внутреннюю память:

– Регистр DP

Поле TY должно быть равно 2 или 3, что указывает на внутреннюю память.

– Регистр DI

Инициализируется начальным адресом блока данных во внутренней памяти куда будут пересылаться данные из порта связи.

Для передачи во внешнюю память:

– Регистр DP

Поле TY должно быть равно 4 или 7, что указывает на внешнюю память.

– Регистр DI

Инициализируется начальным адресом блока данных во внешней памяти, куда будут пересылаться данные из порта связи.

Для выбранного канала в регистре DMACFGH должен быть задан номер устройства с которым работает канал. Биты регистра DMACFGH имеют следующее соответствие: биты 0-3 каналу 8, биты 4-7 каналу 9, биты 8-11 каналу 10, биты 12-15 каналу 11. Передача всегда иницируется устройством. Как только буфер устройства-приемника загружается принятыми данными, посылается запрос к каналу DMA и он выполняет пересылку данных в блок данных описываемый регистром ТСВ.

Каналы DMA 4-11 имеют одну особенность при организации цепочки операций: если в универсальном канале новое значение ТСВ может быть загружено только в регистр того же канала, для каналов 4-11 можно указать номер канала-приемника следующего ТСВ. Правда номер канала-приемника лежит в диапазоне от 4 до 11. Поэтому после окончания работы один канал может иницировать работу другого канала.

### 10.6.2 Внутренняя/внешняя память – устройство-передатчик

Для пересылки данных из внутренней или внешней памяти в устройство должны использоваться только каналы с 4 по 7. Передатчик ТСВ этих каналов должен быть запрограммирован соответствующим образом.

Для передачи из внутренней памяти:

– Регистр DP

Поле TY должно быть равно 2 или 3, что указывает на внутреннюю память.

– Регистр DI

Инициализируется начальным адресом блока данных во внутренней памяти, откуда будут читаться данные для порта связи.

Для передачи из внешней памяти:

– Регистр DP

Поле TY должно быть равно 4 или 7, что указывает на внешнюю память.

- Регистр DI

Инициализируется начальным адресом блока данных во внешней памяти, откуда будут читаться данные для порта связи

Для выбранного канала в регистре DMA CFGL должен быть задан номер устройства, с которым работает канал. Биты регистра DMA CFGL имеют следующее соответствие: биты 16-19 каналу 4, биты 20-23 каналу 5, биты 24-27 каналу 6, биты 28-31 каналу 7.

Передача всегда инициируется устройством-передатчиком. Как только буфер передатчика имеет возможность принять данные - посылается запрос к каналу DMA и он выполняет пересылку данных из блока данных, описываемого регистром TCB, в буфер передатчика.

### 10.6.3 Пересылка между портами связи

Контроллер DMA поддерживает возможность пересылки данных между портами связи без использования промежуточных буферов в памяти. Понятно, что передача может идти от приемника порта связи в передатчик порта связи.

В процессоре имеется два порта связи. Поскольку инициатором передачи в данном случае выступает приемник порта связи, такие передачи поддерживаются только каналами DMA с 8 по 9. Отличием такой пересылки от пересылки в память является необходимость дополнительного контроля готовности передатчика канала связи принять данные от приемника. Особенности программирования TCB приемника каналов с 8 по 9 состоят в следующем:

- Регистр DP

Поле TY должно быть равно 1, что указывает на порт связи.

- Регистр DI

Инициализируется адресом передатчика порта связи, который будет принимать данные. Адрес 0x1F04A0 для порта связи 0, 0x1F04A8 для порта связи 1. На самом деле в данном адресе важны только биты 3 и 4. С их помощью кодируется номер передатчика порта связи. Все другие биты безразличны.

- Модификатор DXM должен быть сброшен.

Передача всегда инициируется готовностью приемника канала связи и готовностью передатчика канала связи. Как только буфер приемника загружается принятыми данными, посылается запрос к каналу DMA. Канал DMA анализирует состояние буфера передатчика, в который он запрограммирован записать данные. Если буфер передатчика канала связи пуст – происходит пересылка.

## 10.7 Канал 12

Канал 12 поддерживает возможность пересылки данных от ведущего устройства во внутреннюю память процессора. В качестве ведущего устройства может выступать только модуль цифрового коррелятора (ЦК). Блок данных в TCB приемника описывается стандартным образом. Параметры работы канала и завершения работы канала определяются стандартным образом. Единственным

ограничением в ТСВ является то, что тип обмена ТУ должен быть равен 2 или 3, т.е. приемник данных всегда внутренняя память.

Канал 12 имеет буфер данных и запрашивает выполнение транзакции, как только в буфере появляются данные. Размер буфера данных составляет четыре 128-разрядных слова. Требованием в работе канала является соответствие размера данных, записываемых в буфер, размеру операнда который определяется полем LEN в регистре ТСВ.

Канал принимает данные в буфер только, когда он активен. Если происходит запись в буфер при неактивном канале – генерируется флаг ошибки.

## 11 Внешний порт

Большинство выводов процессора являются универсальными, т.е. они индивидуально могут быть запрограммированы как вход или выход. Вместе с тем процессор может быть сконфигурирован так, что набор внешних контактов будет образовывать внешний порт, который обеспечивает доступ к внешней памяти, устройствам ввода-вывода, отображаемыми на пространство памяти.

Шина данных может быть сконфигурирована для ширины в 32 или 16 бит. Для достижения высокой тактовой частоты внешней шины, процессор использует конвейерный протокол внешней шины с программируемой глубиной конвейера для доступа к внешней памяти.

### 11.1 Внешняя память

Внешний порт процессора обеспечивает связь с внешней памятью и периферией. Внешняя память и периферия включены в унифицированное адресное пространство процессора. Внутренние шины процессора мультиплексируются на внешний порт, создавая шину внешней системы с одной 22-битной шиной адресов и одной 16\32-битной шиной данных. Внешняя память и устройства могут иметь разрядность 16 или 32 бита. Процессор автоматически упаковывает внешние данные в слова длиной 32, 64 или 128 бит. Встроенное декодирование адресных линий (для генерации сигналов выбора блоков памяти) обеспечивает адресацию устройств внешней памяти. Специальные линии контроля также формируются для упрощения адресации страничного режима динамической памяти.

Процессор использует синхронный конвейер при обменах на внешней шине. Это позволяет осуществлять связь с синхронной DRAM. Опция допускает асинхронную связь для медленных устройств.

Внешние данные могут быть доступны по каналам DMA или через ядро. При доступе посредством ядра задержка чтения может быть значительной, что может вызывать длительные остановы процессора.

Программируемое количество тактов ожидания позволяет работать с периферией имеющей различные требованиями на время доступа.

#### 11.1.1 Интерфейс внешней памяти

Интерфейс внешней памяти позволяет организовать подключение различных типов памяти к процессору. Основными типами памяти являются асинхронная или синхронная статическая память (SRAM), а также синхронная динамическая память (SDRAM).

Контроллер SDRAM расположен на кристалле процессора. Он управляет всеми контрольными сигналами SDRAM (RAS, CAS, SDWE, SDCKE, SDQM) и поддерживает инициализацию и регенерацию микросхем SDRAM. Максимальная производительность SDRAM при передаче данных равна одной передаче за один тактовый цикл. Производительность может удерживаться близко к максимальной, если последовательные доступы выполняются к той же странице

Процессор поддерживает также протокол медленного устройства. Этот протокол следует использовать для устройств не критичных в плане производительности.

Внешняя шина включает в себя шину данных, с пропускной способностью 32 или 16 разрядов, 23-разрядную шину адреса и управляющие сигналы. Процессор всегда ведущий на внешней шине.

В подразделе «Порты общего назначения» отмечалось, что интерфейс внешней памяти реализуется на базе портов общего назначения PxA и PxD. Определяющим конфигурационным регистром является регистр PX\_ALT (Таблица 33). Для реализации полнофункционального внешнего интерфейса (32-разрядная шина данных и 23-разрядная шина адреса) в регистр PX\_ALT необходимо записать значение 0\_0111\_1111b. Это соответствует тому, шина данных и шина адреса полностью контролируются модулем интерфейса внешней памяти. Линии управления внешнего интерфейса используют порт PC и также нуждаются в их конфигурировании. Регистр PC\_ALT позволяет задать необходимую конфигурацию в зависимости от типа используемой памяти.

После того, как модуль внешнего интерфейса получает контроль над внешними выводами микросхемы, он может быть дополнительно сконфигурирован под различные настройки путем записи в конфигурационные регистры SYSCON и SDRCON. Данные регистры задают параметры непосредственно для контроллера внешнего интерфейса. Регистры контроллера внешнего интерфейса доступны как регистры периферийных устройств на SOC-шине. Базовый адрес регистров 0x8000\_0080. Список доступных регистров приведен ниже (Таблица 57).

**Таблица 57 – Группа регистров интерфейса шины**

<b>Имя</b>	<b>Описание</b>	<b>Номер\смещение</b>	<b>Значение после сброса</b>
SYSCON	Регистр конфигурации внешней шины	0x0080 \ 0	0x0000 9067
BUSLOCK	Регистр блокировки внешней шины	0x0083 \ 3	0
SDRCON	Регистр конфигурации SDRAM	0x0084 \ 4	0
SDRSRF	Запрос режима selfrefresh	0x0085 \ 5	0
SYSTAT	Регистр статуса системы; только чтение	0x0086 \ 6	0
SYSTATCL	Регистр статуса системы; адрес для сброса ошибок. только чтение	0x0087 \ 7	0
BMAX	16-разрядный регистр общего назначения	0x008C \ 0xc	0

#### **11.1.1.1 Особенности внешней шины**

- Ширина шины данных: 16 или 32 разряда;
- Конвейерные передачи с программируемым числом этапов конвейера;
- Программируемые состояния простоя (IDLE);
- Протокол поддерживает циклы ожидания, добавляемые ведомым устройством с использованием вывода ACK;
- Интерфейс EPROM и FLASH: 8-разрядная шина данных с фиксированным числом циклов ожидания, поддержкой чтения и записи;
- SDRAM-интерфейс;
- Поддержка медленных устройств ввода/вывода;
- Поддержка передач через DMA для внешних устройств ввода/вывода в режиме квитирования.

### 11.1.1.2 Внешние контакты ввода/вывода интерфейса шины

Ниже представлено описание основных выводов внешнего интерфейса, используемых для реализации обмена со всеми типами памяти. Буква «n» в начале названия вывода указывает на низкий активный уровень сигнала.

– **ADDR22-0**

Шина адреса. Выдает на эти линии адрес устройства, к которому выполняется доступ.

– **DATA31-0**

Шина данных с тремя состояниями. Процессор выдает данные на эти выводы или принимает данные или команды с помощью этих выводов. Имеется возможность задания размера шины (16 или 32 бита) для работы с различными устройствами системы.

– **nRD**

Чтение памяти. nRD активируется всякий раз, когда процессор осуществляет чтение из подчиненного устройства системы. Активный уровень низкий. Не используется при доступе к SDRAM.

– **nWR**

Запись слова. nWR активируется в случае, когда происходит запись во внешнее устройство. nWR изменяется одновременно с выводами ADDR. Активный уровень низкий. Не используется при доступе к SDRAM.

– **ACK**

Подтверждение готовности. Процессор анализирует вход ACK во время доступа к устройствам. Внешние подчиненные устройства могут деактивировать (установить в низкий уровень) ACK чтобы добавить состояние ожидания во время доступа к ним. ACK используется подчиненными устройствами на фазе данных. Высокий уровень на линии ACK означает готовность устройства, низкий – требование подождать. ACK не используется при доступе к SDRAM.

– **nBMS**

Выбор загрузочной памяти. nBMS активизируется, когда процессор выполняет доступ к загрузочной памяти (EPROM или флэш). nBMS изменяется одновременно с выводами ADDR. Активный уровень низкий.

– **nMS1-0**

Выбор памяти. nMS0 или nMS1 активируются в то время, когда процессор желает получить доступ к банкам памяти 0 или 1 соответственно. Выходы nMS1-0 являются декодированными сигналами адреса памяти и изменяются параллельно с выводами ADDR. Когда линии ADDR31-27 равны 0b00110, nMS0 активирован. Когда линии ADDR31-27 равны 0b00111, nMS1 активирован. Активный уровень низкий.

– **MSSD3-0**

Выборка памяти SDRAM. MSSD0, MSSD1, MSSD2 или MSSD3 активны (один из четырех), всякий раз, когда процессор выполняет доступ к пространству памяти SDRAM. MSSD3-0 являются выводами декодированного адреса

памяти и активны в то время, когда процессор выполняет командный цикл SDRAM. Некоторые команды активируют сразу все четыре линии.

– **nRAS**

Выборка адреса строки. Во время чтения или записи SDRAM низкий уровень сигнала RAS указывает на то, что на шине адреса находится достоверный адрес строки. В остальных случаях доступа к SDRAM он определяет тип операции для выполнения в соответствии со спецификацией SDRAM.

– **nCAS**

Выборка адреса столбца. Во время чтения или записи SDRAM низкий уровень сигнала CAS указывает на то, что на шине адреса находится достоверный адрес столбца. В остальных случаях доступа к SDRAM он определяет тип операции для выполнения в соответствии со спецификацией SDRAM.

– **DQM**

Маска слова данных SDRAM. При высоком уровне сигнала переводит буферы данных DQ SDRAM в третье состояние. При чтении всегда равен нулю. При операциях записи DQM является активным (равным 1), для запрещения записи слова.

– **SDA10**

Вывод разряда 10 адреса SDRAM. SDRAM использует отдельный вывод адресной линии номер 10 из-за того, что она используется в специальных операциях и может задействоваться в то время, когда процессор выполняет на внешней шине транзакцию не связанную с SDRAM.

– **SDCKE**

Разрешение тактового сигнала SDRAM. Вход синхронизации микросхемы динамической памяти может быть постоянно активным, но микросхема будет управляться данным синхросигналом, только если на линии SDCKE высокий уровень. Низкий уровень сигнала SDCKE используется для перевода памяти в режим саморегенерации.

– **nSDWE**

Разрешение записи SDRAM. При низком уровне сигнала во время активного CAS, SDWE указывает на выполнение записи SDRAM. При высоком уровне сигнала во время активного CAS, SDWE указывает на выполнение чтения SDRAM. В остальных операциях SDRAM, SDWE определяет тип операции для выполнения в соответствии со спецификацией SDRAM.

### **11.1.1.3 Структура интерфейса внешней памяти**

Передаваемые на внешней шине данные могут быть стандартными словами (32 разряда), двойными словами (64 разряда) или счетверенными словами (128 разрядов). Адрес этих слов должен быть выровнен по их размеру. Поскольку ширина шины (16 или 32 разряда) может быть меньше размера передачи, передача данных распределяется по нескольким циклам внешней шины.

Внешний порт (EBIU) всегда является мастером на внешней шине, а также мастером и подчиненным на внутренних шинах процессора. Рисунок 12 показывает структуру внутренних шин, соединяющих внешний порт с внутренними модулями процессора. Два главных элемента передачи данных в структуре внешнего порта:

- выходное FIFO (OFIFO) передачи запросов на внешнюю шину;

- входное FIFO (IFIFO) входных запросов с внешней шины или возвращаемых данных с внешней шины.

Буфер IFIFO используется в случае передачи данных во внутренний приемник в ответ на чтение ядром процессора или контроллером DMA устройств на внешней шине. В этой ситуации внешний интерфейс работает как мастер на внешней шине и возвращает прочитанные данные. Далее данные отправляются на SOC-шину (контроллер DMA или устройства) или в ядро процессора (в IFIFO SOC-интерфейса).

Выходной буфер OFIFO внешнего порта используется для всех транзакций направленных во внешнее адресное пространство. Это могут быть запросы процессорного ядра или контроллера DMA.

**Потоки данных, которые имеют место при различных вариантах передач, представлены ниже**

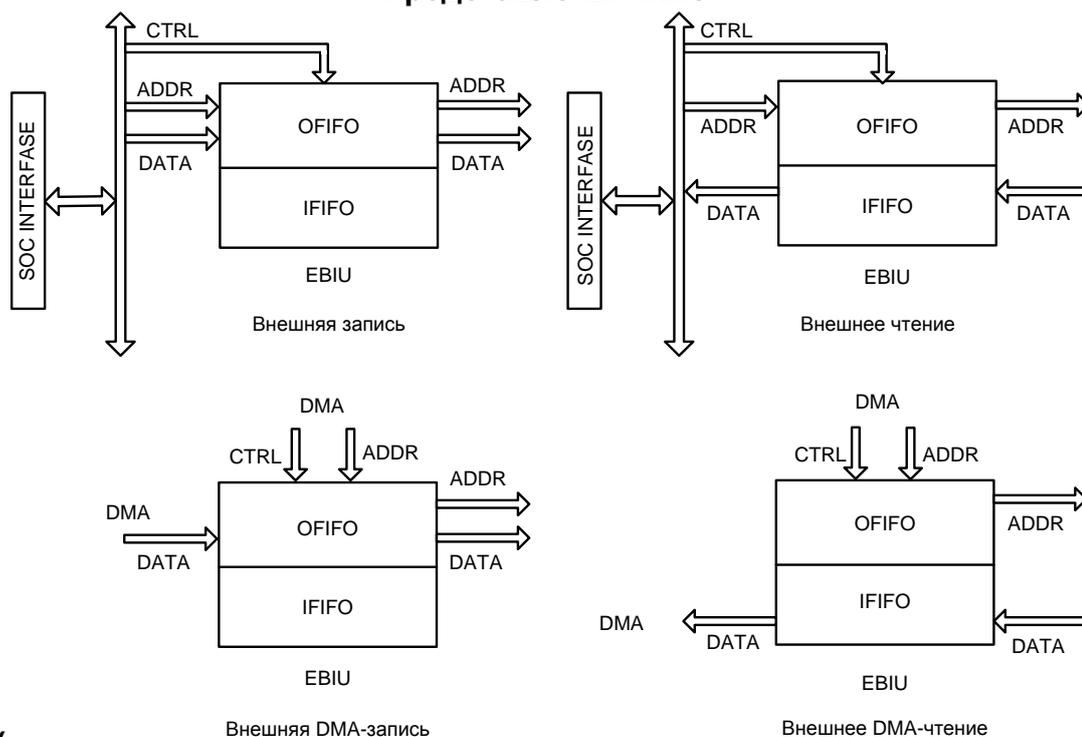


Рисунок 16). Возможны операции чтения и операции записи. Операции, выполняемые ядром или контроллером DMA.

Если операция является операцией записи, она завершается:

- для ядра процессора в момент записи данных в SOC OFIFO;
- для контроллера DMA в момент записи в EBIU OFIFO.

Если выполняется операция чтения, данные проходят следующие этапы:

- При чтении ядром процессора, запрос помещается в SOC OFIFO, далее он перемещается в EBIU OFIFO, где читается и анализируется внешним интерфейсом. Выполняется цикл чтения внешнего устройства и прочитанные данные размещает в EBIU IFIFO, откуда они пересылаются в SOC IFIFO. SOC интерфейс принятые данные отправляет в регистр назначения.
- При чтении контроллером DMA, запрос помещается в EBIU OFIFO, где затем читается и анализируется внешним интерфейсом. Выполняется цикл чтения внешнего устройства и прочитанные данные размещает в EBIU

IFIFO, откуда они пересылаются в буфер контроллера или буфера передатчиков устройств.

При доступе к ресурсам со стороны многих устройств используется приоритет доступа. EBIU IFIFO участвует в доступе к SOC-шине и к SOC-IFIFO. При доступе к SOC-шине приоритет следующий:

- SOC OBUF (наивысший);
- EBIU IFIFO;
- процессорное ядро (низший).

При доступе к SOC IFIFO приоритет следующий:

- чтение ядром регистра SOC шины;
- EBIU IFIFO;
- контроллер ДМА.

Приоритет устройств фиксированный и не зависит от приоритета транзакции ДМА или приоритета транзакции внешнего мастера.

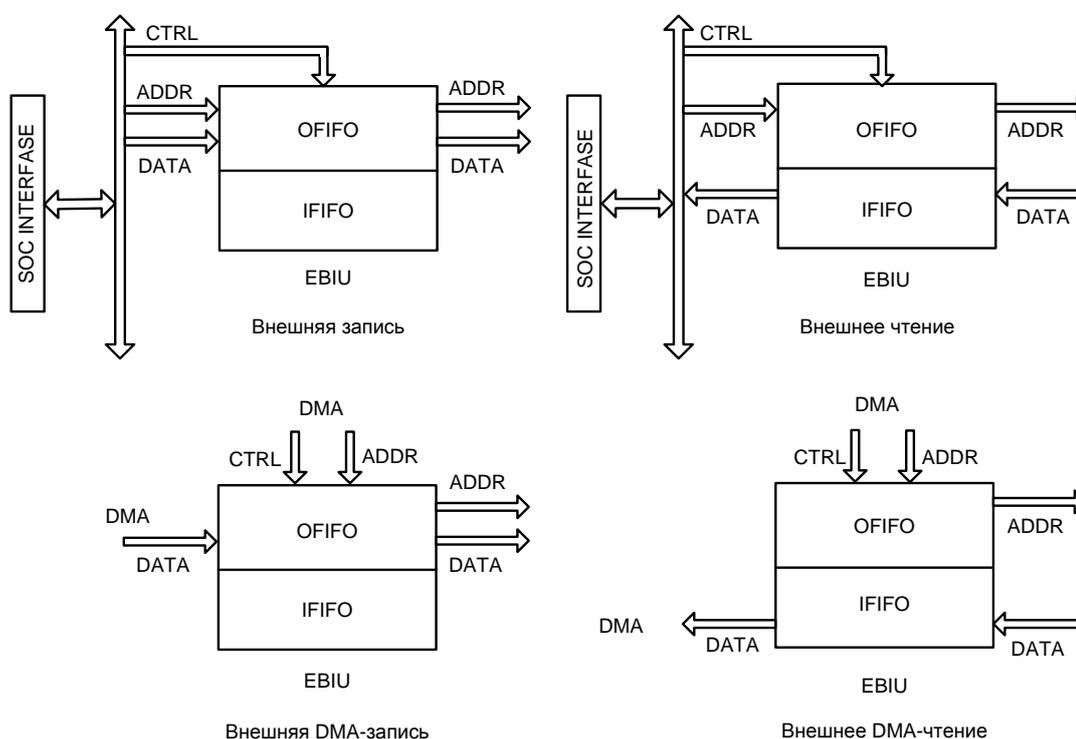


Рисунок 16 – Передача данных через интерфейс внешней шины (EBIU)

#### 11.1.1.4 Программирование регистра SYSCON

Регистр SYSCON является регистром конфигурации контроллера внешнего интерфейса и должен быть запрограммирован до того, как начнется передача данных на шине (если значения по умолчанию должны изменяться). В режиме пользователя доступ к SYSCON невозможен. Описание разрядов регистра приведено ниже (Таблица 58). Рассмотрим подробно назначение различных разрядов регистра.

Таблица 58 – Регистр SYSCON

Бит	Имя	Назначение
-----	-----	------------

0	BNK0IDLE	Вставка пустого цикла между операциями доступа к банку памяти MS0: 1 – вставить цикл; 0 – нет
2:1	BNK0WAIT	Количество внутренних циклов ожидания при обращении к банку MS0: 00 – 0 циклов 01 – 1 цикл 10 – 2 цикла 11 – 3 цикла
4:3	BNK0PIPE	Глубина конвейера банка памяти MS0: 00 – 1 цикл 01 – 2 цикла 10 – 3 цикла 11 – 4 цикла
5	BNK0SLOW	Тип протокола обмена для банка MS0: 0 – медленный (асинхронный) 1 – синхронный (конвейерный)
6	BNK1IDLE	Вставка пустого цикла между операциями доступа к банку памяти MS1: 1 – вставить цикл 0 – нет
8:7	BNK1WAIT	Количество внутренних циклов ожидания при обращении к банку MS1: 00 – 0 циклов 01 – 1 цикл 10 – 2 цикла 11 – 3 цикла
10:9	BNK1PIPE	Глубина конвейера банка памяти MS1: 00 – 1 цикл 01 – 2 цикла 10 – 3 цикла 11 – 4 цикла
11	BNK1SLOW	Тип протокола обмена для банка MS1: 0 – медленный (асинхронный) 1 – синхронный (конвейерный)
12	HOSTIDLE	Вставка пустого цикла между операциями доступа к области памяти хоста: 1 – вставить цикл 0 – нет
14:13	HOSTWAIT	Количество внутренних циклов ожидания при обращении к области хоста: 00 – 0 циклов 01 – 1 цикл 10 – 2 цикла 11 – 3 цикла
16:15	HOSTPIPE	Глубина конвейера для области хоста: 00 – 1 цикл 01 – 2 цикла 10 – 3 цикла 11 – 4 цикла
17	HOSTSLOW	Тип протокола обмена для области хоста: 0 – медленный (асинхронный) 1 – синхронный (конвейерный)
18	-	
19	MEMWIDTH	Ширина шины данных при обращении к внешней памяти 1 – 16 бит 0 – 32 бита

20		
21		
23:22		Всегда 0
24		Режим работы входного ФИФО внешнего интерфейса 0 – нормальный режим 1 – ускорение передачи данных Бит можно только установить. Сброс только аппаратно.
25		Режим работы выходного буфера внешнего интерфейса 0 – нормальный режим 1 – ускорение передачи данных Бит можно только установить. Сброс только аппаратно
26		Всегда 0
27	XSDM	Модификация адресного пространства. 0 – стандартное адресное пространство. 1 – модифицированное адресное пространство в котором адреса банков 0010b и 0001b соответствуют адресному пространству динамической памяти. Используется для поддержки байтовой адресации внешней памяти.
31:28		Всегда 0

### **Ширина шины данных**

Физически шина данных внешнего интерфейса имеет 32 линии. Однако это не значит, что подключаемая память или внешнее устройство обязательно должны иметь такую же разрядность шины данных. С помощью регистра PX\_ALT можно задать ширину шины данных равной 32, 16 или 8 бит. Контроллер внешнего интерфейса будет управлять тем количеством выводов шины данных, значение которых определено в регистре PX\_ALT. Вместе с этим сам контроллер поддерживает алгоритм работы с 32-разрядной или 16-разрядной внешней шиной данных. Это означает, что если в регистре SYSCON[19] задана разрядность шины данных 32 бита, то, несмотря на установки в регистре PX\_ALT, интерфейс будет вести себя так, как будто бы шина имеет разрядность 32 бита. Если в регистре SYSCON[19] установлена разрядность шины данных 16 бит, все словные операции интерфейс будет превращать в две последовательные операции по 16 бит.

### **Протокол медленного устройства для шины**

Для настройки протокола медленного устройства обнуляется разряд протокола медленного устройства SLOW (разряд 5 для банка 0, разряд 11 для банка 1 и разряд 17 для хоста). В случае выбора протокола медленного устройства некоторые поля (поле глубины конвейера PIPE, разряд IDLE) являются безразличными и не используются. Поле ожидания определяет число внутренних циклов ожидания при медленных доступах. Если число внутренних ожиданий равно нулю, механизм внешних ожиданий (анализ входа ACK) не может использоваться для данных транзакций.

### **Конвейерный протокол для шины**

Для выбора конвейерного протокола бит SLOW соответствующего банка памяти устанавливается в 1, а поле PIPE устанавливается на заданную глубину конвейера (0b00 для одного цикла, 0b01 для двух циклов, 0b10 для трех циклов и 0b11 для четырех циклов). Так определяется глубина конвейера только для транзакций чтения. Глубина конвейера для транзакций записи всегда равна одному циклу. Разряд IDLE устанавливается, если на шине есть несколько подчиненных устройств в одном банке памяти, с целью предотвращения конфликта на шине

данных между различными подчиненными устройствами при последовательном чтении. Если банк памяти содержит только одно подчиненное устройство, разряд IDLE сбрасывается. Поле внутреннего ожидания безразлично при конвейерных транзакциях.

#### **Начальные значения для работы шины**

После сброса регистр SYSCON имеет начальное значение равное 0x9067. Пользователь может не менять данное значение, если оно соответствует конфигурации его внешней шины. Начальное значение определяет:

- банк 0: медленный протокол, три состояния ожидания, включен холостой режим;
- банк 1: конвейерный протокол, конвейер глубиной 1 цикл, нет состояний ожидания, включен холостой режим;
- хост: конвейерный протокол, конвейер глубиной два цикла, нет состояний ожидания, включен холостой режим.

#### **11.1.1.5 Интерфейс конвейерного протокола**

Процессор использует конвейерный протокол для соединения с быстрой синхронной памятью, подключаемой к nMS1-0. Конвейерный протокол обеспечивает передачу данных со скоростью частоты внешней шины. При конвейерном доступе адрес и управляющие разряды выдаются одновременно в одном такте, а данные лишь спустя несколько циклов: от одного до четырех в зависимости от направления передачи и установленных параметров. Процессор может выдавать адрес следующей транзакции еще до прихода данных предыдущей транзакции.

Управляющие сигналы, используемые в конвейерных транзакциях:

- **nRD** – если активен указывает на транзакцию чтения;
- **nWR** – транзакция записи, если один из этих сигналов активный;
- **ACK** – управляется адресуемым подчиненным устройством в течение цикла данных. Если имеет высокий уровень, подчиненное устройство готово завершить цикл данных, в противном случае мастер шины должен ждать готовности подчиненного устройства.

#### **Базовая конвейерная транзакция**

Временная диаграмма базовой конвейерной транзакции представлена ниже (Рисунок 17). Во время адресного цикла, адрес передается вместе с сигналами RD или WR.

Цикл данных может начинаться с задержкой от одного до четырех тактов, в соответствии с параметрами, указанным в регистре конфигурации для подчиненного устройства, и направлением транзакции. Задержка между циклом адреса и циклом данных есть глубина конвейера. В примере глубина конвейера равна двум (Рисунок 17). В цикле данных данные передаются в соответствии с направлением транзакции (чтение или запись). Если подчиненное устройство готово к завершению транзакции, оно активирует сигнал ACK (равен 1) и принимает или выставляет данные. Если подчиненное устройство не готово, оно деактивирует сигнал ACK (равен 0) и задерживает данные.

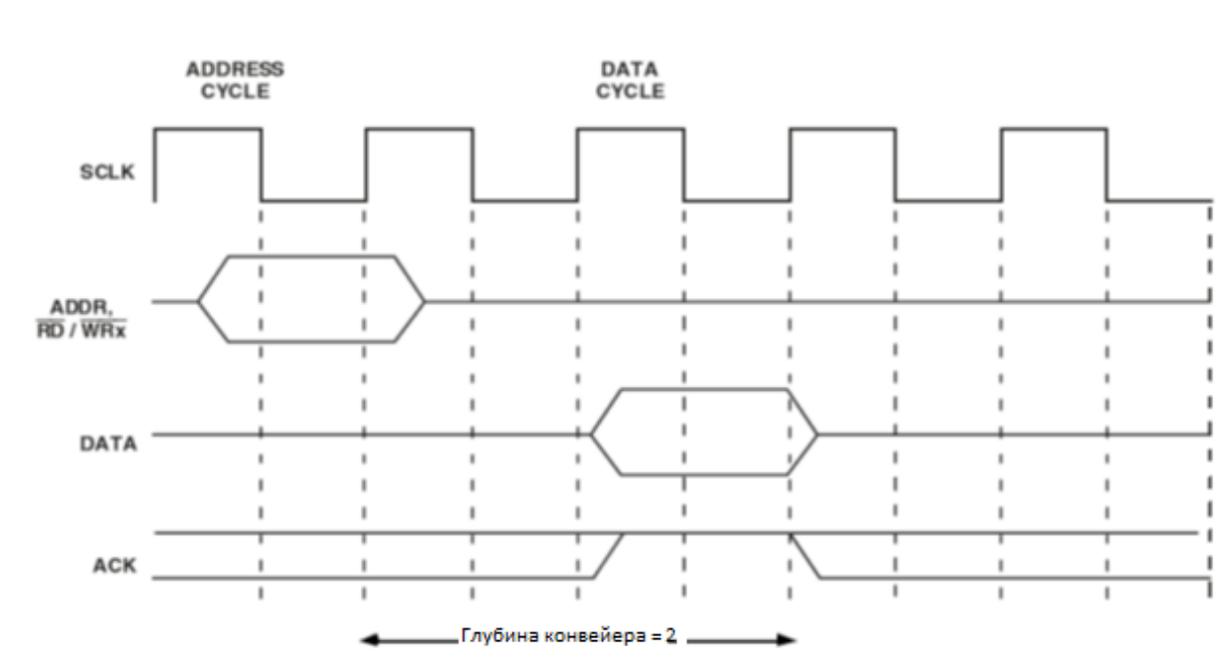


Рисунок 17 – Основные конвейерные передачи

Глубина конвейера программируется для каждого из банков памяти, а также зависит от типа транзакции (чтение или запись). Для записи глубина конвейера всегда равна 1, т.е. данные выставляются процессором на шину в следующем такте после выдачи адреса. При чтении из внешних банков памяти глубина конвейера программируется пользователем и может быть до четырех циклов. Глубина конвейера может быть выбрана индивидуально для каждого из банков в регистре SYSCON после сброса.

Одиночные транзакции занимают различное число циклов в зависимости от размера транзакции и ширины внешней шины:

- транзакция стандартного слова: один цикл;
- транзакция двойного слова на 32-разрядной шине: 2 цикла;
- транзакция двойного слова на 16-разрядной шине: 4 цикла;
- транзакция счетверенного слова на 32-разрядной шине: 4 цикла;
- транзакция счетверенного слова на 16-разрядной шине: 8 циклов;

Тип транзакции определяется сигналами RD и WR во время цикла адреса.

#### Циклы ожидания

В транзакциях чтения, если подчиненное устройство вовремя готово к циклу данных выполняемой транзакции, подчиненное устройство активирует сигнал ACK в цикле данных. Если подчиненное устройство не готово, оно деактивирует сигнал ACK в цикле данных и держит его неактивным (равным 0) пока не будет готово продолжить транзакцию. ACK также может быть деактивирован после транзакции записи, если подчиненное устройство не может вовремя отправить данные в место назначения.

Сигнал ACK может быть выставлен в любом цикле данных в многоцикловых транзакциях. Нет никаких ограничений на продолжительность удерживания ACK в нуле.

Рисунок 18 демонстрирует использование сигнала ACK. В этом примере, данные DA1 не готовы вовремя и выставляются на один такт позже. Сигнал ACK

деактивируется во время цикла данных DA1 и активируется в следующем цикле, чтобы указать на достоверные данные. Из-за этого цикла ожидания:

- мастер берет с шины данные на один цикл позже;
- адресный цикл, следующий за сигналом АСК (AB2) растягивается на один цикл;
- все подчиненные устройства с приостановленными транзакциями во время цикла, следующего за сигналом АСК, замедляются на число циклов, которые были деактивированы сигналом АСК. Например, между адресным циклом АВ0 и циклом данных DV0 проходит пять циклов, хотя глубина конвейера равна только четырем.

При этом адресные и управляющие сигналы на линиях в цикле, где АСК был в нуле, должны защелкиваться подчиненными устройствами.

Транзакции записи используют циклы ожидания АСК не так, как транзакции чтения. В случае записи подчиненное устройство активирует сигнал АСК, пока у него имеется свободное место в буфере приема.

Ниже представлен пример, в котором адресный цикл AA3 заставляет подчиненное устройство деактивировать сигнал АСК (Рисунок 19). Это может быть вызвано тем, что подчиненное устройство приняло 128-разрядное слово в буфер, и он занят для следующей транзакции. Остановка удлинняет конвейер транзакции на два цикла. Циклы, которые были остановлены АСК, начинаются спустя один такт после деактивации сигнала АСК – например, АВ2 и DV1 дополнены двумя циклами.

При этом все подчиненные устройства принимают адрес АВ1, а устройство, которому адресованы данные DV0, также принимает эти данные во внутренний буфер.

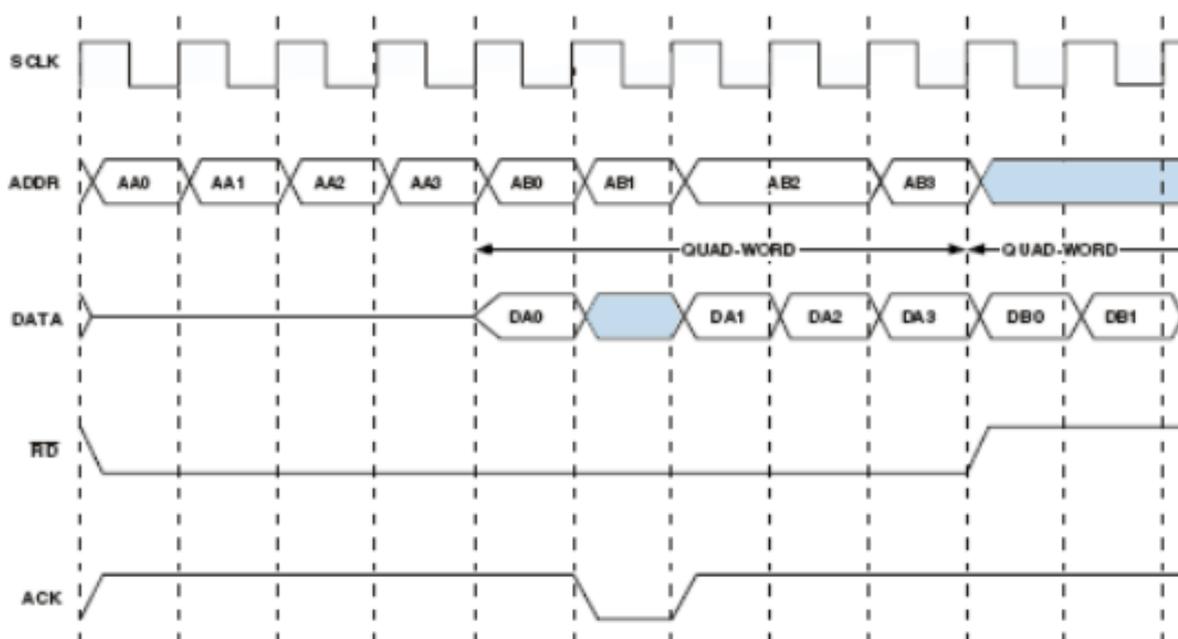


Рисунок 18 – Доступ к пакетному чтению с последующим пакетным чтением

Транзакции АВ0 и АВ1 адресуют другое подчиненное устройство (Рисунок 19). Если первое подчиненное устройство переводит сигнал АСК на низкий уровень и следующие транзакции соответствуют другому подчиненному устройству, новое

подчиненное устройство транзакции не может управлять сигналом АСК до тех пор, пока первое подчиненное устройство не завершит транзакцию.

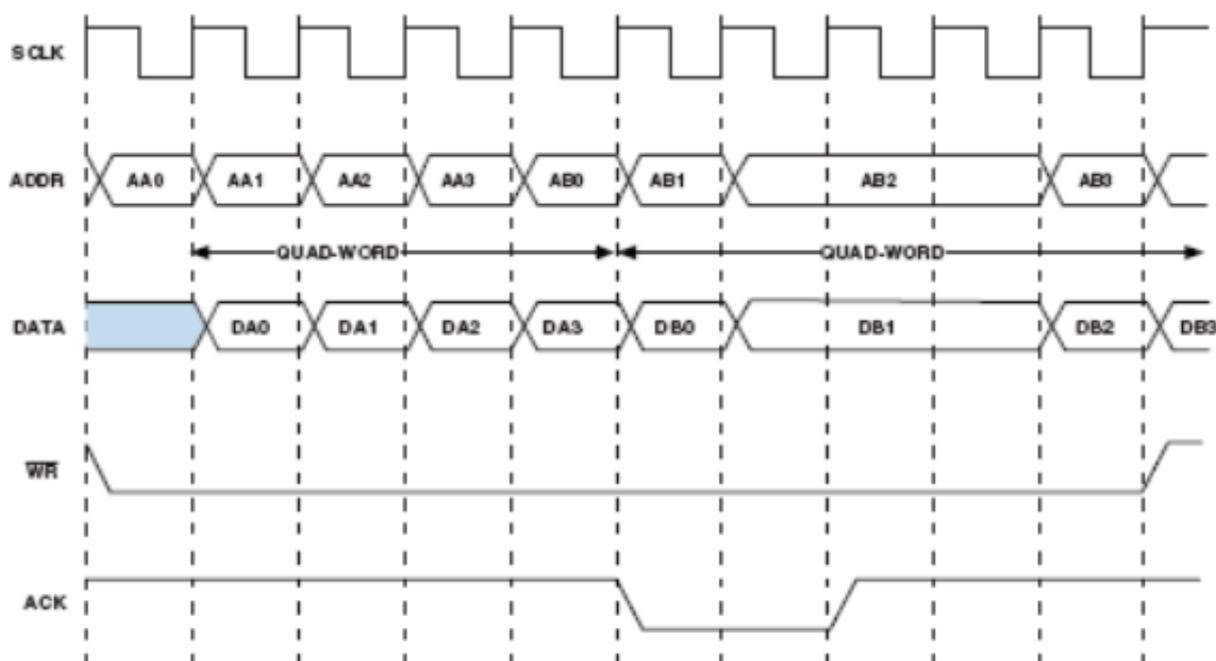


Рисунок 19 – Доступ к пакетной записи с последующей пакетной записью

### 11.1.1.6 Интерфейс протокола медленного устройства

Процессор поддерживает протокол медленных устройств, который может быть использован для простых устройств. Этот протокол может быть сконфигурирован для адресных пространств принадлежащих банку 0 или банку 1. Протокол устанавливается посредством программирования соответствующих разрядов в регистре SYSCON:

- бит SLOW равен 1;
- биты PIPE глубины конвейера равны нулю;
- циклы ожидания WAIT программируются в соответствии с требованиями системы;
- цикл IDLE (холостой цикл) равен 1.

Целью таких установок является прямое соединение с простыми, медленными, некритическими модулями памяти или периферийными устройствами. Протокол может работать с синхронными и асинхронными устройствами. Базовый протокол с конфигурацией «без циклов ожидания» представлен ниже (Рисунок 20, Рисунок 21).

В первом цикле выдается адрес и активируется линия выбора соответствующего банка памяти.

Во втором цикле активируются сигналы RD или WR (в зависимости от типа транзакции). Если транзакция записи, в этом цикле процессор выдает данные. В случае транзакции чтения в этом цикле подчиненное устройство начинает выдавать данные. В случае чтения, в конце цикла процессор защелкивает данные с шины во внутреннем регистре.

В третьем цикле процессор продолжает удерживать адрес и данные (если выполняется запись), но деактивирует сигналы RD или WR. В случае чтения устройство перестает выдавать данные на шину.

Для конфигурации «без циклов ожидания», циклы ожидания не могут быть добавлены деактивирующим сигналом ACK.

Одно из ключевых требований при организации протокола медленного устройства – гарантированное время удержания адреса и данных. Из временной диаграммы видно, что процессор обеспечивает один такт предустановки адреса и один такт удержания адреса и данных. Дополнительные такты предустановки данных могут обеспечиваться внутренними программируемыми циклами ожидания или внешним сигналом ACK.

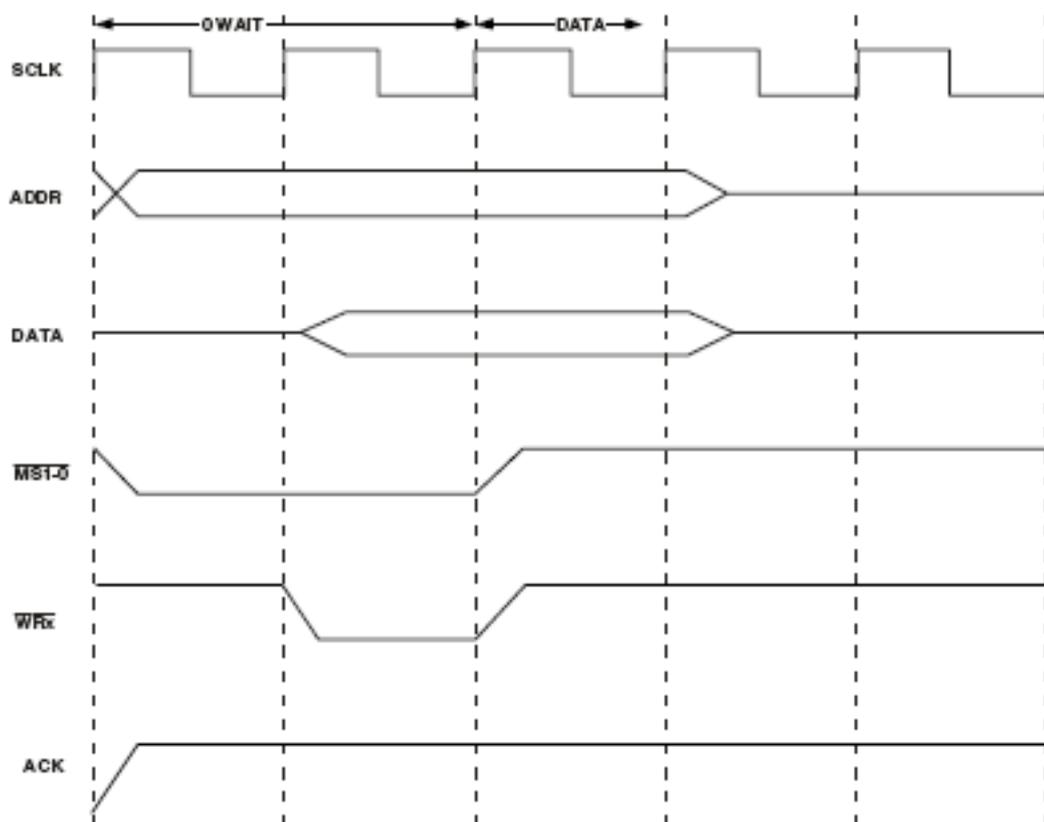


Рисунок 20 – Медленный протокол. Запись с нулевым временем ожидания

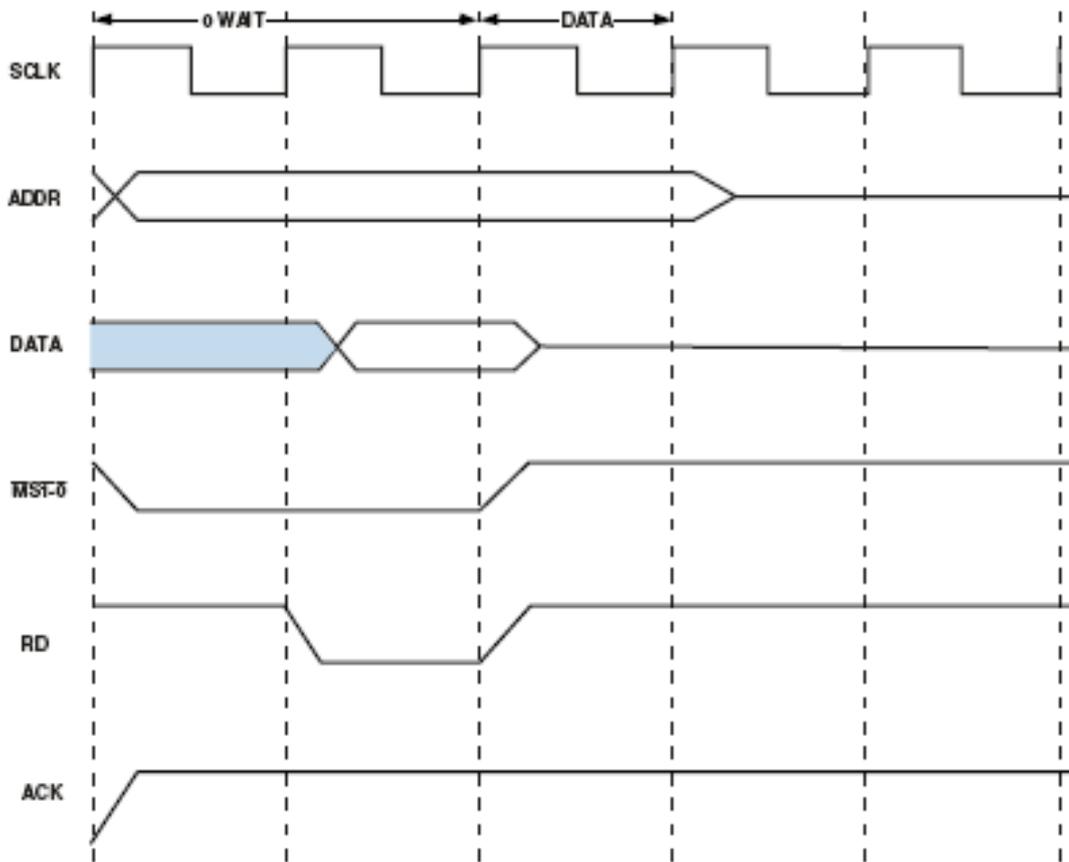


Рисунок 21 – Медленный протокол. Чтение с нулевым временем ожидания

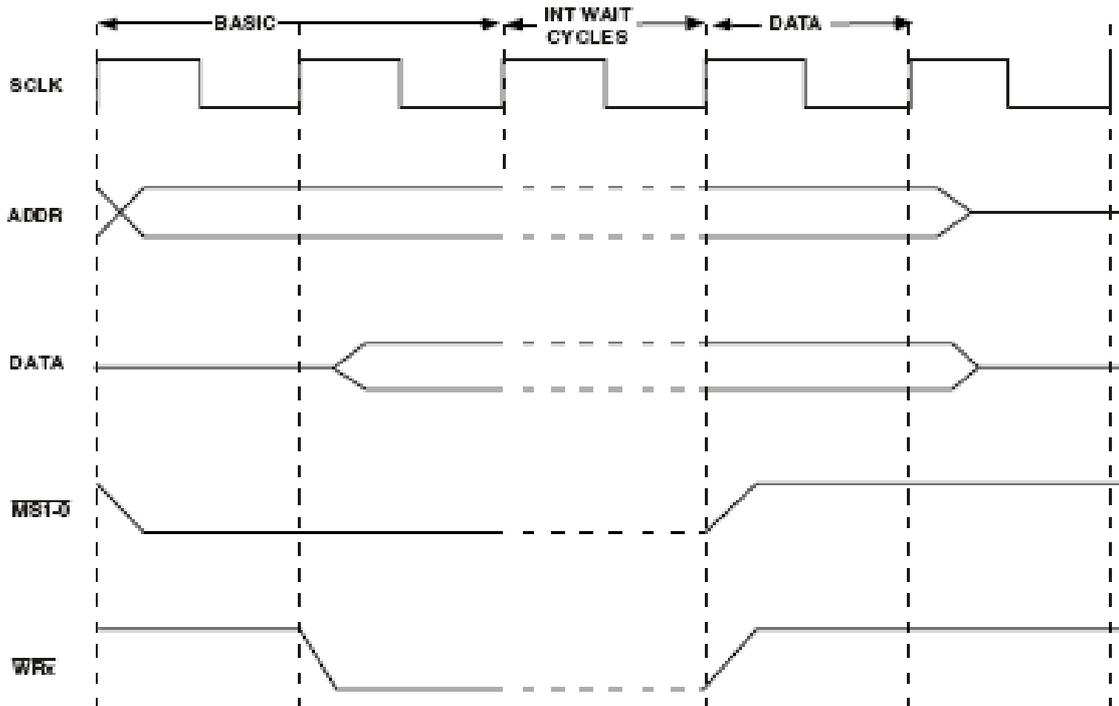


Рисунок 22 – Медленный протокол. Запись с циклами ожидания

В протоколе медленного устройства при необходимости могут быть вставлены циклы ожидания. Регистр SYSCON включает в себя поле задания некоторого числа внутренних циклов ожидания. Значения могут быть от нуля до трех.

Рисунок 22 и Рисунок 23 показывают медленные транзакции с одним или более циклами ожидания. Ситуация похожа на транзакцию без циклов ожидания, но второй цикл (цикл в котором активированы RD или WR) повторяется в соответствии с числом циклов внутреннего ожидания.

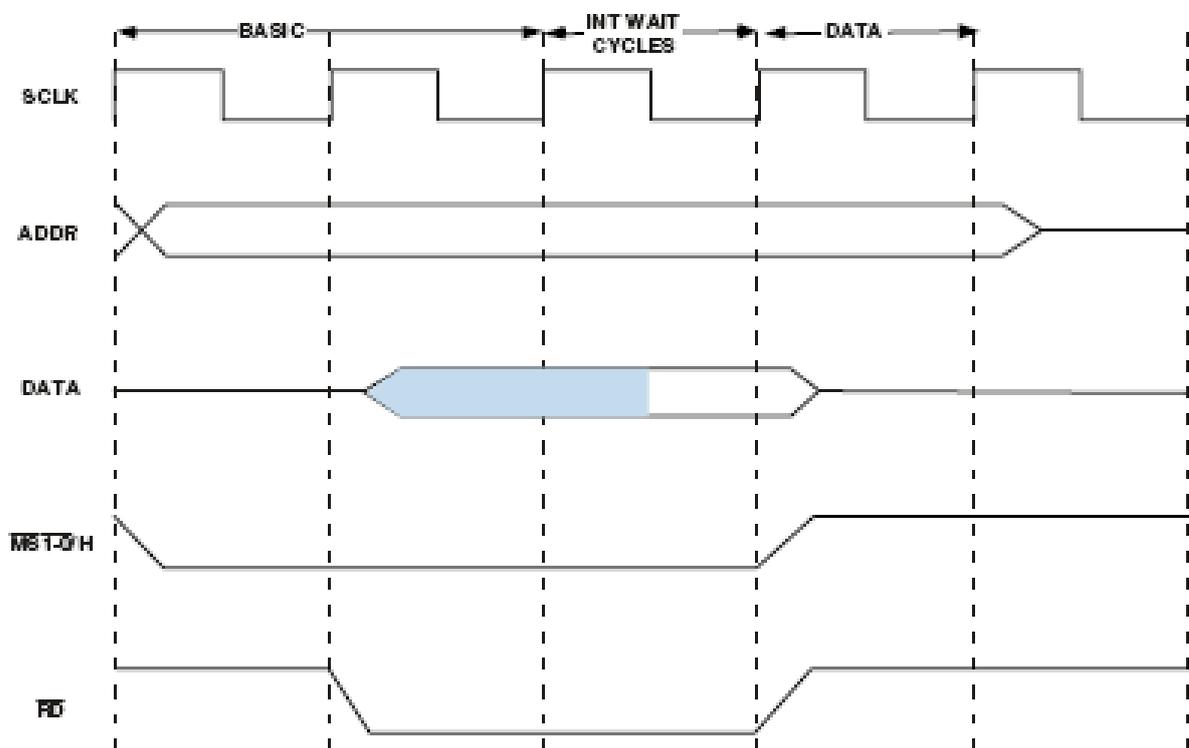


Рисунок 23 – Медленный протокол. Чтение с циклами ожидания

Дополнительные циклы внешнего ожидания могут быть вставлены путем деактивации сигнала ACK. Это может быть сделано только тогда, когда в регистре SYSCON поле WAIT не равно нулю. Для того, чтобы добавить циклы внешнего ожидания, сигнал ACK должен быть деактивирован в течение одного или более циклов перед последним циклом ожидания (см. Рисунок 24).

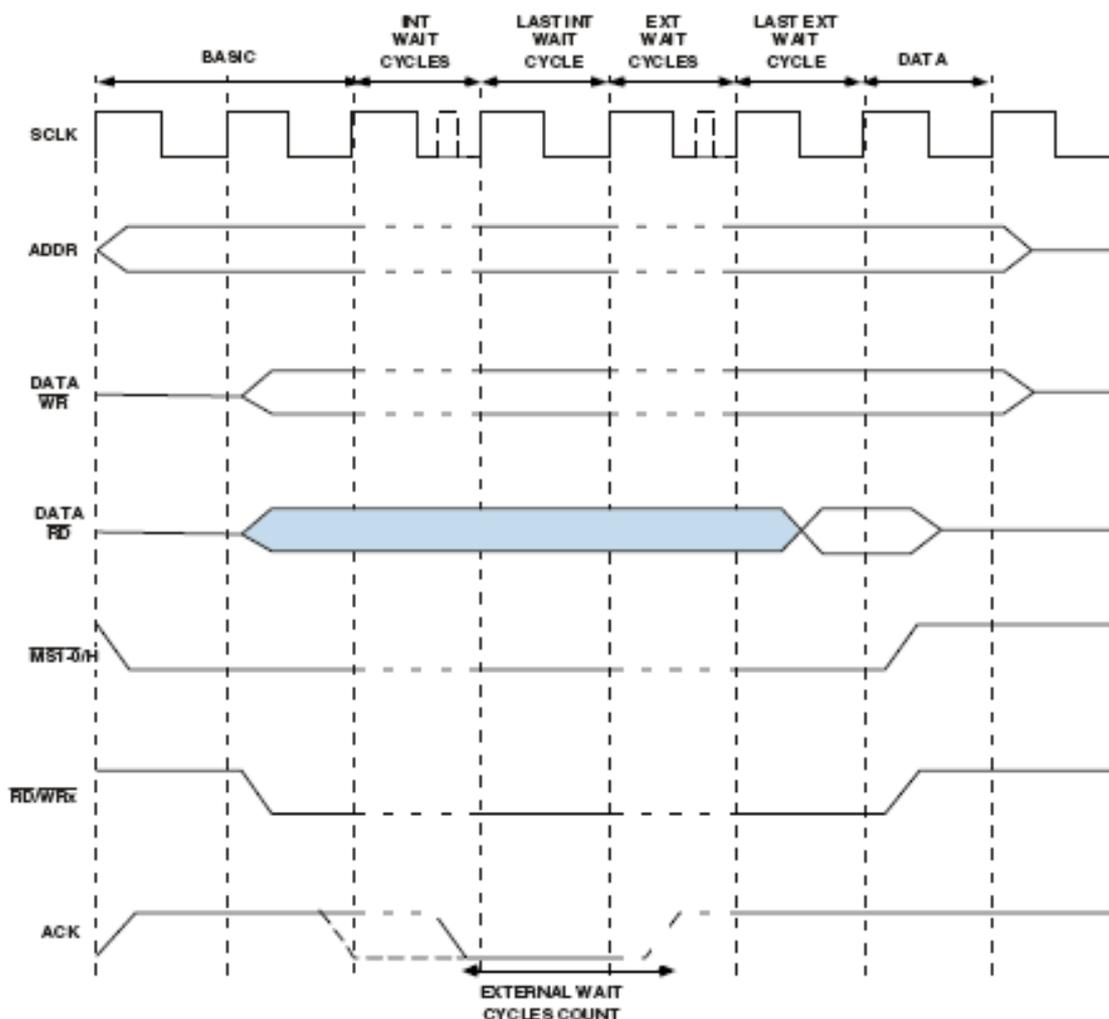


Рисунок 24 – Медленный протокол. Запись/чтение с внешними циклами ожидания

### 11.1.1.7 Интерфейс EPROM

Процессор имеет возможность работать с внешней 8-разрядной перепрограммируемой памятью типа EPROM. Во время сброса процессор может быть сконфигурирован для загрузки из внешнего EPROM. В этом случае программа загружается из EPROM во внутреннюю память автоматически как часть последовательности сброса. Для этой цели используется канал 0 DMA.

Поскольку разрядность шины данных памяти EPROM составляет 8 бит, внешний интерфейс процессора упаковывает принимаемые байты в 32-разрядные данные.

Процессор может работать с EPROM не только во время сброса. Имеется возможность доступа и в процессе выполнения программы. Однако в работе с EPROM имеется одна особенность – память EPROM не является частью адресного пространства процессора. Процессор поддерживает словную адресацию внешней памяти, а память EPROM имеет байтовую организацию. В связи с этим для работы с EPROM можно использовать только канал общего назначения ДМА. Для работы с EPROM тип обмена в TCB канала устанавливается равным 6. При чтении EPROM, обмен может происходить только между EPROM и внутренней памятью. Внешний интерфейс определяет, что текущий запрос есть запрос ДМА и его тип равен 6. Это позволяет интерфейсу выбрать для обмена протокол медленного устройства с фиксированным числом циклов внутреннего ожидания равным 16. При выполнении

внешней транзакции процессор активизирует линию BMS для выбора EPROM. Для передачи данных используются биты с 0 по 7-й шины данных.

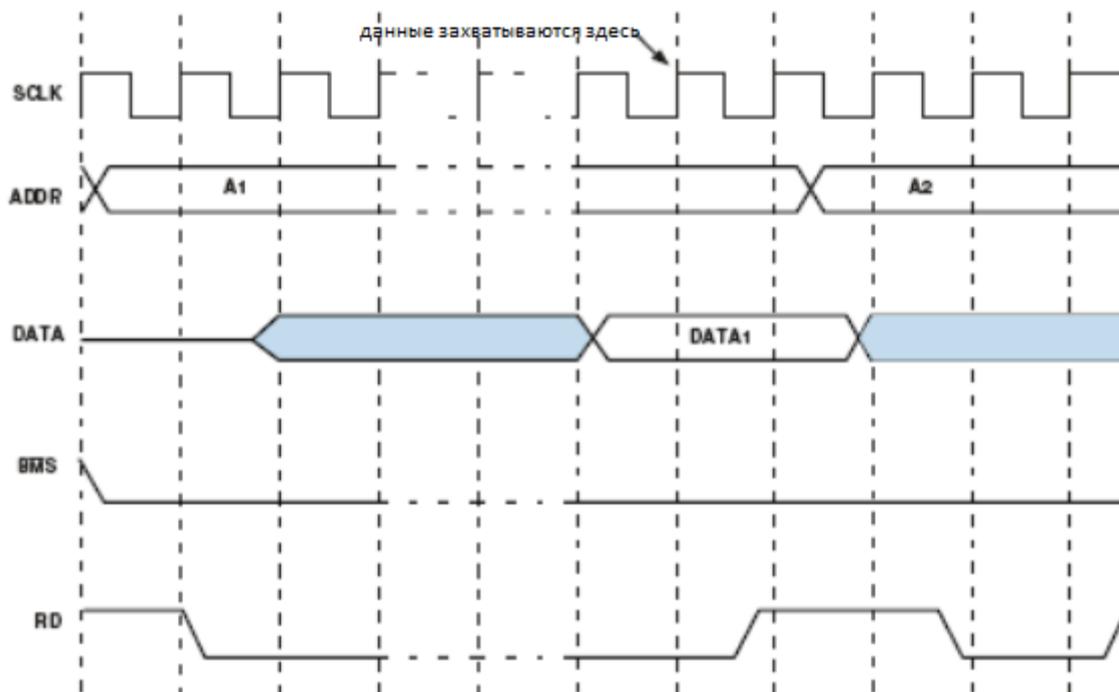


Рисунок 25 – Доступ к загрузке EPROM – 16 циклов ожидания

Возможно как чтение внешней EPROM, так и операция записи для программирования памяти. Во время чтения интерфейс упаковывает байты в слова и передает их во внутреннюю память. Операция записи идентична чтению, но с одним отличием – при записи нет распаковки слова в байты. Процессор может выдавать на внешнюю шину только 32-разрядные слова, а EPROM может принимать только младший байт. Поэтому при записи программист сам должен подготовить информацию в удобном для записи виде. При записи EPROM источником данных может быть как внутренняя, так и внешняя память.

Во внутреннем или внешнем ОЗУ процессора данные должны быть организованы так, как показано ниже (Рисунок 26). Слева исходные данные для программирования. Справа показаны распакованные в слова байты для передачи во внешнюю EPROM.

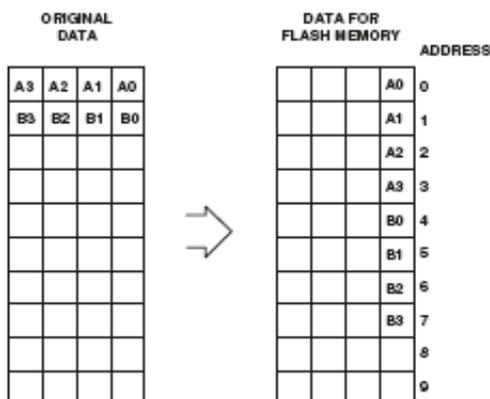


Рисунок 26 – Подготовка данных для записи в 8-разрядную EPROM

При выполнении обмена с EPROM, внешний интерфейс всегда устанавливается размер шины данных 32 разряда, несмотря на установки в SYSCON. При этом адрес EPROM не должен быть в используемом адресном пространстве хоста, SDRAM или банков MS1 и MS0. Лучше всего старшие биты адреса установить в ноль. При чтении внешний интерфейс автоматически устанавливает размер данных равным квадрослову. Это позволяет произвести четыре цикла чтения, взять из них по одному байту и сформировать одно слово. Канал ДМА при этом должен выполнять обмен словами, но увеличивать адрес на 4. Таким образом, первое считывание будет по адресам 0, 1, 2, 3, а второе по адресам 4, 5, 6, 7. Такая небольшая хитрость используется при чтении.

При записи используется размер данных, который задает канал ДМА. По сути при записи внешний интерфейс работает с EPROM так как будто эта память имеет разрядность 32 бита и передает ей слова данных. Основной хитростью здесь является предварительная подготовка данных для записи в памяти.

#### **11.1.1.8 Интерфейс SDRAM**

Процессор имеет выделенное адресное пространство для подключения синхронной динамической памяти SDRAM. Четыре банка в этом адресном пространстве выбираются линиями выборки MSSD3–0. SDRAM доступно в адресном диапазоне с 0x4000 0000 до 0x7FFF FFFF.

Процессор поддерживает стандартную (3,3 В) и пониженной мощности (2,5 В) SDRAM. Память работает с использованием частоты внешней шины SCLK (системная частота). Все входы защелкиваются и все выходы корректны при положительном (из 0 в 1) перепаде частоты. Синхронный интерфейс позволяет выполнять передачу данных каждый такт. SDRAM поддерживает несколько типов пакетных доступов в зависимости от инициализации регистра режима SDRCON. Каждый тип доступа предваряется запуском соответствующей команды в SDRAM. Специальные режимы SDRAM должны инициализироваться в регистре SDRCON.

SDRAM имеет внутреннюю организацию в виде двух или четырех банков. Выводы выбора банка SDRAM определяют, к какому банку происходит обращение. SDRAM имеет программируемый параметр задержки чтения, который должен инициализироваться приложением в соответствии с типом устройства и рабочей тактовой частотой.

Для соответствия требуемым временным характеристикам SDRAM, процессор может выполнять конвейерную передачу адреса и управляющих команд SDRAM. Для этого используется бит глубины конвейера в регистре SDRCON.

Микросхемы SDRAM поставляются различными производителями. Каждый поставщик имеет свои временные требования касательно параметров задержки команд ACT - PRE (tRAS) и PRE - ACT (tRP). Для поддержки всех основных поставщиков и различных уровней скорости, регистр SDRCON программируем для того, чтобы разработчик мог удовлетворить временные требования микросхемы SDRAM.

Таковыми параметрами являются битовые поля: CAS задержка, PRE - RAS задержка, RAS - PRE задержка.

Наилучшая скорость обмена с SDRAM получается при использовании блочного обмена, когда процессор выполняет несколько последовательных чтений или записей в один банк памяти. Интерфейс процессора имеет возможность отслеживания и поддержки пакетного доступа.

Особенностью работы динамической памяти является необходимость регенерации всех ее ячеек памяти в течение заданного интервала времени. Это гарантирует сохранность информации. Процессор имеет возможность

программировать частоту регенерации для соответствия временным требованиям микросхемы.

При описании интерфейса SDRAM используются следующие определения:

- Команда активации банка АСТ  
Активирует выбранный банк и фиксирует в нем новую строку. Должна использоваться перед командой чтения или записи.
- Длина пакета (Burst Length)  
Определяет количество слов, которые поступают на вход или выход SDRAM после детектирования команды чтения или записи. Микросхема памяти всегда программируется для длины пакета равного полной странице.
- Тип пакетирования (Burst Type)  
Определяет порядок, в котором SDRAM отправляет или принимает пакетные данные после детектирования команды чтения или записи. Процессор поддерживает только последовательный доступ.
- Задержка CAS (CAS Latency)  
Задержка, в тактах, между тем, когда SDRAM определяет команду чтения и когда данные поступают на выходные выводы. Величина запаздывания определяется уровнем скорости устройства и тактовой частотой шины. Приложение должно программировать этот параметр в регистре SDRCON.
- Маскирование ввода-вывода DQM данных  
Вывод DQM используется для маскирования контроллером операций записи.
- Регистр SDRCON  
Регистр, который содержит программируемые конфигурационные параметры для возможности работы контроллера SDRAM с микросхемой SDRAM.
- Регистр режима (Mode register)  
Регистр конфигурации SDRAM, который содержит определяемые пользователем параметры, согласованные с регистром SDRCON. Этот регистр находится внутри микросхемы памяти.
- Размер страницы (Page Size)  
Процессор поддерживает 1024-, 512-, и 256-словные размеры страниц. Размер страницы может быть запрограммирован в регистре SDRCON.
- Команда подзарядки (Precharge)  
Подзаряжает активный банк.
- Период регенерации (Refresh Rate)  
Программируемая величина в регистре SDRCON. Период регенерации позволяет приложениям координировать скорость SCLK с требуемой частотой регенерации SDRAM.
- Саморегенерация (Self-Refresh)  
Состояние микросхемы памяти, в котором она использует внутренний таймер и периодически инициирует автоматические регенерации памяти

без внешних команд. Этот режим работы обеспечивает SDRAM режим низкого потребления.

- tRAS  
Требуемая задержка между запуском команды активации строки банка и запуском команды подзарядки. Величина зависит от производителя и задается в регистре SDRCON.
- tRC  
Требуемая задержка между последовательными командами активации одного банка. Данный параметр зависит от производителя и определяется как  $tRC = tRP + tRAS$ . Процессор фиксирует значение этого параметра, так что это непрограммируемая опция.
- tRCD. RAS - CAS задержка. Требуемая задержка между командой ACT и началом первой операции записи или чтения. Данный параметр зависит от производителя и определяется как  $tRCD = CL$ . Процессор использует фиксированное значение этого параметра, так что это непрограммируемая опция.
- tRP  
Требуемая задержка между запуском команды подзарядки и командой активации. Данный параметр зависит от производителя и определяется в SDRCON.

Рисунок 27 показывает интерфейс контроллера SDRAM между внутренним ядром процессора и внешним устройством SDRAM.

Процессор обычно генерирует адрес внешней памяти, который затем активизирует соответствующую линию выбора SDRAM памяти (MSSD3–0), а также указывает на тип операции обмена: чтение или запись. Эта информация анализируется контроллером SDRAM. Внутренняя 32-битная шина адреса мультиплексируется контроллером SDRAM для создания соответствующего сигнала выбора памяти, адреса строки, адреса колонки и банка в SDRAM.

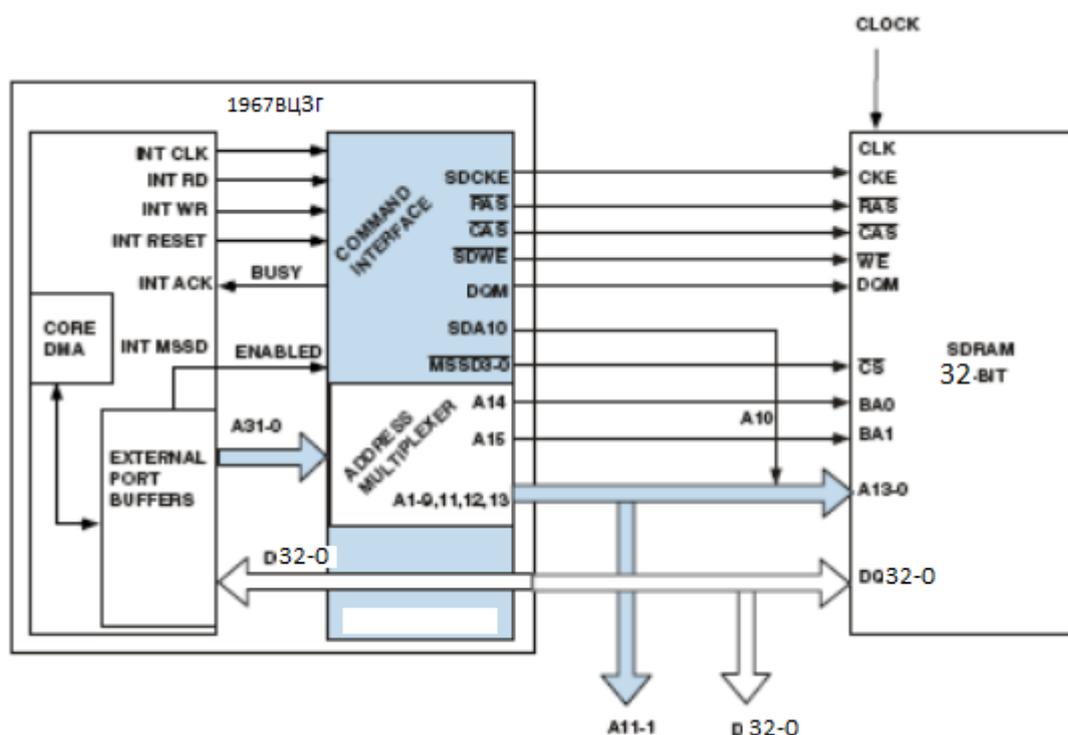


Рисунок 27 – Интерфейс контроллера SDRAM (для конфигурации с 32-разрядной шиной)

#### Контакты ввода/вывода интерфейса SDRAM

Интерфейс SDRAM порта внешней шины использует для обмена с SDRAM памятью общую шину данных и часть шины адреса. Для управления микросхемой памяти используются дополнительные линии:

- **MSSD3–0;**
- **RAS;**
- **CAS;**
- **DQM;**
- **SDA10;**
- **SDCKE;**
- **SDWE.**

Назначение линий было описано ранее. Выводы данных и адреса SDRAM могут соединяться напрямую с выводами данных и адреса процессора. Ширина шины данных может быть выбрана 16 или 32 разряда при программировании регистра SYSCON.

Подключения линий адреса:

- разряды 9-0 адреса SDRAM соединяются с выводами ADDR9–0 процессора;
- разряд 10 адреса SDRAM соединяется с выводом SDA10 процессора;
- разряды 15-11 адреса SDRAM соединяются с выводами ADDR15–11 (столько разрядов сколько требуется) процессора.

#### Выводы выбора банка

Внутренняя организация микросхемы динамической памяти включает в себя два или четыре банка. Для выбора банка микросхема имеет специальные выводы BS1-0. Эти выводы подключаются к адресным линиям процессора. Соединение

выводов адреса процессора в качестве линий выбора банка для устройства SDRAM меняется в зависимости от следующих факторов:

- рабочее напряжение, используемое для SDRAM (стандартно 3,3 В или 2,5 В в SDRAM с пониженной мощностью);
- число банков.

Для двух банков BS равен ADDR11 или любой адресной линии в диапазоне ADDR15–11. Для четырех банков BS1–0 равен ADDR12–11 или BS1–0 может быть равен другой паре адресов в диапазоне ADDR15–11.

Для SDRAM с пониженной мощностью использование линий адресов в качестве сигналов выбора банка ограничено ADDR15-14. Для двух банков может быть использована любая из двух адресных линий ADDR14 или ADDR15. Для четырех банков должны использоваться ADDR15–14.

**Физическое соединение внутреннего адреса процессора и адреса микросхемы SDRAM**

Выше отмечалось, что SDRAM может подключаться к шине данных шириной 16 бит или 32 бита. Выбор подключения зависит от объема памяти, которую пользователь хочет установить в системе, и от желаемой скорости обмена.

Таблица 59, Таблица 60 и Таблица 61 описывают выравнивание данных внешнего порта и соединения SDRAM для 32-разрядной системной шины.

В таблицах приведено соответствие между внешними контактами адреса процессора и контактами адреса микросхемы SDRAM, а также указано какие разряды внутреннего 32-битного адреса выдаются на эти контакты в разных фазах доступа (т.е. в фазе активации банка и выбора активного ряда (строки)), а также в фазе чтения или записи при выборе колонки (номера слова в строке).

При 16-разрядной шине чтение (RD) и запись (WR) для передачи данных всегда используются два такта: сначала младшая часть слова, а затем старшая. Основным отличием при 16-разрядной шине данных при выдаче адреса является то, что 32-разрядный внутренний адрес сдвигается влево на один разряд перед выдачей на внешние контакты. Самый младший бит адреса в первом такте обмена всегда равен нулю, а во втором такте равен 1.

**Таблица 59 – Размер страницы 256 слов, 32-разрядная шина**

<b>Физический вывод процессора Процессор</b>	<b>Внутренний адрес активного цикла банка</b>	<b>Внутренний адрес периода выборки столбца</b>	<b>Физический вывод SDRAM</b>
A0	8	0	A0
A1	9	1	A1
A2	10	2	A2
A3	11	3	A3
A4	12	4	A4
A5	13	5	A5
A6	14	6	A6
A7	15	7	A7
A8	16	8	A8
A9	17	9	A9
A10	несущественный	несущественный	NC
SDA10	18	0	A10/AP
A11	19	19	A11 или банк

A12	20	20	A12 или банк
A13	21	21	A13 или банк
A14	22	22	A14 или банк
A15	23	23	A15 или банк

**Таблица 60 – Размер страницы 512 слов, 32-разрядная шина**

<b>Физический вывод процессора Процессор</b>	<b>Внутренний адрес активного цикла банка</b>	<b>Внутренний адрес периода выборки столбца</b>	<b>Физический вывод SDRAM</b>
A0	9	0	A0
A1	10	1	A1
A2	11	2	A2
A3	12	3	A3
A4	13	4	A4
A5	14	5	A5
A6	15	6	A6
A7	16	7	A7
A8	17	8	A8
A9	18	9	A9
A10	несущественный	несущественный	NC
SDA10	19	нулевой	A10/AP
A11	20	20	A11 или банк
A12	21	21	A12 или банк
A13	22	22	A13 или банк
A14	23	23	A14 или банк
A15	24	24	A15 или банк

**Таблица 61 – Размер страницы 1 К слов, 32-разрядная шина**

<b>Физический вывод процессора Процессор</b>	<b>Внутренний адрес активного цикла банка</b>	<b>Внутренний адрес периода выборки столбца</b>	<b>Физический вывод SDRAM</b>
A0	10	0	A0
A1	11	1	A1
A2	12	2	A2
A3	13	3	A3
A4	14	4	A4
A5	15	5	A5
A6	16	6	A6
A7	17	7	A7
A8	18	8	A8
A9	19	9	A9
A10	несущественный	несущественный	NC
SDA10	20	нулевой	A10/AP
A11	21	21	A11 или банк
A12	22	22	A12 или банк
A13	23	23	A13 или банк
A14	24	24	A14 или банк
A15	25	25	A15 или банк

**Программирование параметров SDRAM**

Микросхемы SDRAM могут быть получены от разных поставщиков и в зависимости от поставщика, микросхемы могут иметь различные требования к последовательности включения и временным параметрам. С целью возможности работы с большим количеством поставщиков, в процессоре предусмотрен регистр SDRCON в котором можно запрограммировать некоторые параметры SDRAM. Описание разрядов регистра приведено ниже (Таблица 62).

**Таблица 62 – Регистр SDRCON**

<b>Бит</b>	<b>Имя</b>	<b>Назначение</b>
0	SDREN	Включение контроллера SDRAM 1 – включен 0 – выключен
2:1	CAS	Задержка CAS 00 -1 цикл 01 – 2 цикла 10 – 3 цикла 11 – резерв
3	PIPE	Дополнительный конвейер: 1 – используется 0 – нет
5:4	PAGE	Размер страницы: 00 – 256 слов 01 – 512 слов 10 – 1024 слов 11 – резерв
6	-	
8:7	REF	Период регенерации памяти (в тактах SCLK) 00 – 1100 циклов 01 – 1850 циклов 10 – 2200 циклов 11 – 3700 циклов
10:9	PRC2RAS	Задержка от подзаряда до строба RAS (такты SCLK) 00 – 2 01 – 3 10 – 4 11 – 5
13:11	RAS2PRC	Задержка от строба RAS до команды подзаряда: 000 – 2 ..... 110 – 8
14	INIT	Выбор последовательности инициализации: 1 – команда MRS после регенерации памяти 0 – команда MRS перед регенерацией памяти
15	EMREN	Использование дополнительного регистра режима: 1 – разрешено 0 – не используется
31:16	-	Всегда 0

Регистр SDRCON хранит информацию о конфигурации интерфейса SDRAM. Ниже подробно описывается назначение каждого из параметров регистра.

### **Включение контроллера SDRAM**

Разряд SDREN должен быть установлен, если в системе есть SDRAM. В противном случае, он должен быть сброшен. Любой доступ к SDRAM, пока этот разряд сброшен, вызывает аппаратное прерывание ошибки доступа. Установка данного бита включает в работу контроллер SDRAM внешнего интерфейса. При включении контроллер сразу же начинает процедуру инициализации внешней динамической памяти

### **Выбор значения задержки CAS Latency**

Значение задержки CAS определяет задержку в тактовых циклах системы (SCLK), между временем, когда SDRAM обнаруживает команду чтения и временем когда он выдает данные на его внешние выводы. Этот параметр позволяет процессору знать, в каком такте после передачи в SDRAM команды чтения он может принять прочитанные данные.

Задержка CAS не используется в циклах записи.

### **Опция буферизации SDRAM. Глубина конвейера**

Ранее были описаны ситуации, при которых пользователю может понадобиться использовать буферизацию адресных и управляющих сигналов. Связано это с максимальной нагрузочной способностью выводов процессора и требуемой скоростью обмена. В случае использования буферизации в конвейере чтения данных появляется дополнительный цикл. Чтобы данный цикл был учтен контроллером SDRAM и используется бит PIPE. Шина данных не буферизируется. Поэтому при установленном разряде PIPE (1), контроллер SDRAM задерживает данные в цикле записи в течение одного такта.

В случае чтения контроллер SDRAM осуществляет захват данных на один цикл позже.

Ниже приведен пример одного процессора, в котором интерфейс SDRAM соединяется с множеством банков SDRAM (Рисунок 28). Микросхема SDRAM имеет шину данных шириной 4 бита. Для обеспечения 32-разрядной шины данных используется восемь микросхем. Чтобы уменьшить нагрузку на внешние выводы процессора используется буферный регистр который состоит из двух регистров А и В, каждый из которых запоминает одинаковую информацию, но передает ее на свои четыре микросхемы памяти

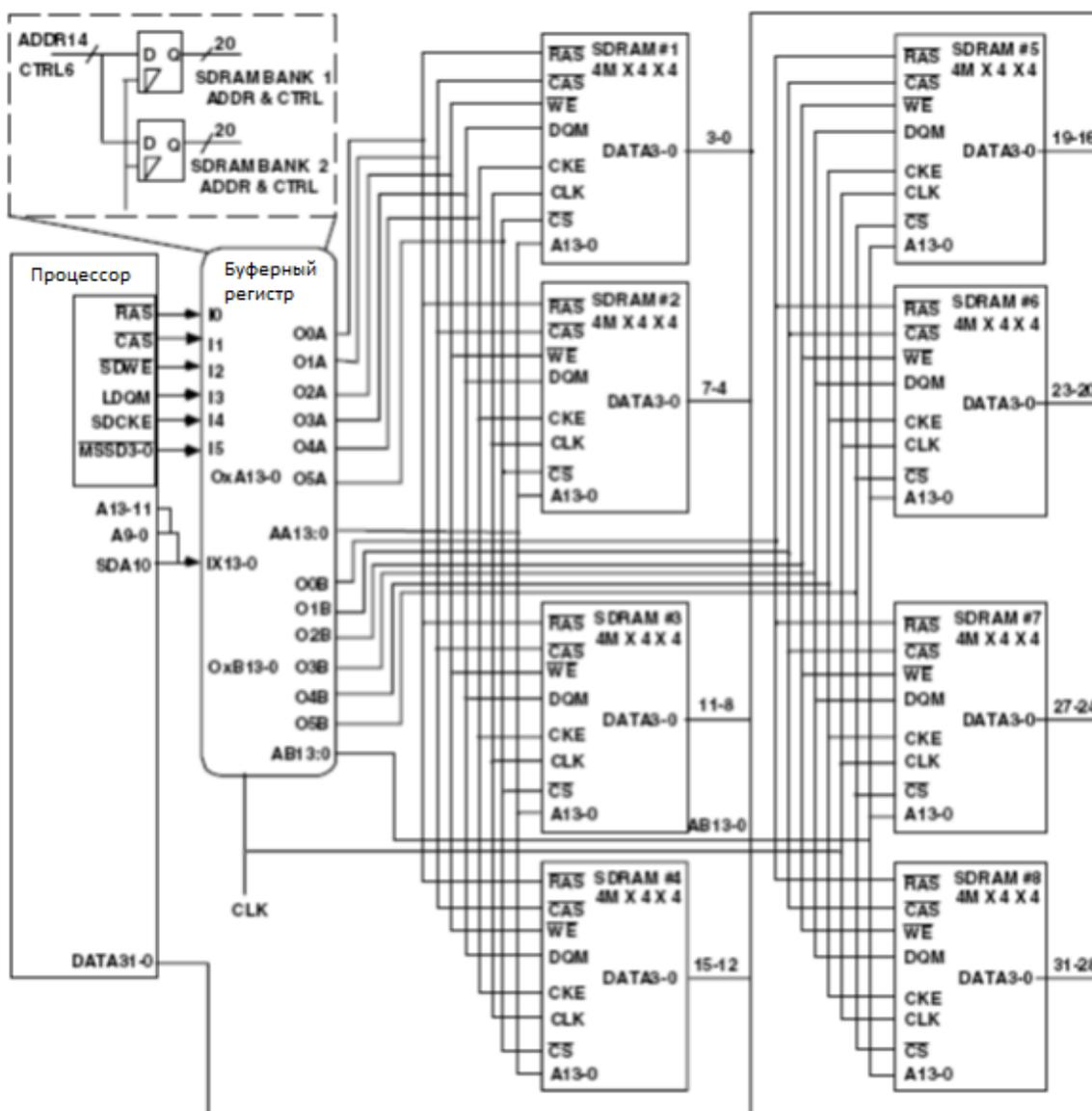


Рисунок 28 – Однопроцессорная система со стандартными устройствами SDRAM (32-разрядная шина)

### Выбор размера страницы SDRAM

Размер страницы определяется конкретным типом микросхемы SDRAM. Разряды PAGE определяют размер страницы банков SDRAM, в количестве слов. Знание размера страницы необходимо контроллеру SDRAM для отслеживания ситуаций обращения к одной и той же странице. Это позволяет существенно увеличить скорость обмена. Также размер страницы влияет на распределение разрядов внутреннего адреса между адресами строки, столбца и выбора банка памяти.

### Период регенерации памяти

Память SDRAM требует, чтобы в течение определенного интервала времени все ее ячейки были обновлены (регенерированы). Это гарантирует сохранность информации. В связи с этим контроллер SDRAM должен периодически посылать в микросхемы памяти команды регенерации. Период подачи команд программируется. Период внутри контроллера задается **в тактах клока внешней шины**.

Количество тактов в периоде рассчитывается исходя из следующего уравнения.

$$f_{cycles} = \left( sdramclock \times \frac{t_{ref}}{rows} \right) = (clock \times refresh\_rate)$$

Период регенерации всей памяти  $T_{ref}$  делится на количество рядов (строк) в одном банке памяти. Получаем период для регенерации одного ряда. Далее это значение умножается на частоту динамической памяти, и мы получаем период подачи команды регенерации в тактах динамической памяти. Также можно получить этот же период в тактах SOC шины. В регистре конфигурации можно задать несколько фиксированных значений периода.

#### **Задержка от подзаряда до строба RAS**

Протокол обмена с динамической памятью требует, чтобы при доступе к новому ряду банка или перед выполнением регенерации памяти, выполнялся подзаряд (закрытие) текущей активной строки. Для подзаряда используется команда **PRE**. После выполнения подзаряда невозможно сразу перейти к активизации следующего ряда т.к. спецификация микросхемы требует, чтобы между подзарядом и активизацией нового ряда была обеспечена некоторая минимальная задержка. Значение этой задержки, выраженное числом системных тактовых циклов (SCLK), и программируется в поле PRC2RAS.

#### **Выбор задержки между RAS и предварительным зарядом**

Аналогично предыдущему параметру, существует требование минимального интервала времени между подачей команды RAS (активизация ряда банка) и подачей команды PRE подзаряда банка. Данный интервал задается числом системных тактовых циклов SCLK. Значение поля RAS2PRC может изменяться от 0 до 6, это соответствует числу циклов SCLK от 2 до 8. Если, например, поле RAS2PRC равно 011 это означает, что минимальный интервал равен пяти циклам. Это не означает, что после команды RAS минимум через пять тактов должна быть команда PRE. Это означает, что если процессор выполнил команду RAS, а затем по каким-то причинам (например, регенерация) ему необходимо выполнить подзаряд, процессор должен проверить прошло ли с момента команды RAS как минимум пять тактов. Тогда он может подавать команду подзаряда.

#### **Выбор последовательности инициализации SDRAM**

Для того чтобы микросхема SDRAM корректно работала, она должна быть предварительно проинициализирована. Процессор обеспечивает две наиболее часто используемые последовательности инициализации.

Бит последовательности инициализации INIT в регистре SDRCON выбирает режим включения SDRAM. Когда бит установлен (1), контроллер SDRAM последовательно генерирует: команду PRE, восемь циклов регенерации и команду MRS (установка регистра режима). Когда бит INIT сброшен (0), контроллер SDRAM работает в следующем порядке: команда PRE, команда MRS и восемь циклов регенерации.

#### **Разрешение регистра расширенного режима (EMR)**

Бит EMREN в регистре SDRCON разрешает программирование регистра расширенного режима. Когда бит установлен (1), контроллер SDRAM формирует цикл

EMRS, предшествующий команде MRS. Когда бит сброшен (0), последовательность инициализации происходит так, как описано в главе выше.

Бит разрешающий EMR должен быть установлен только во время обмена данными с устройствами SDRAM пониженной мощности (2,5 В). Именно в таких микросхемах используется дополнительный регистр режима.

#### **Включение после сброса**

После сброса содержимое регистра SDRCON равно нулю и контроллер SDRAM выключен. Как только приложение пользователя выполняет запись в регистр SDRCON данных с установленным битом включения контроллера SDRAM, контроллер тут же иницирует выбранную последовательность включения динамической памяти. Последовательность инициализации определяется разрядом INIT и разрядом регистра расширенного режима в регистре SDRCON. Программный сброс не вызывает сброса контроллера SDRAM и не иницирует повторно последовательность включения.

### **11.1.2 Команды контроллера SDRAM**

Этот раздел описывает каждую команду, которую контроллер SDRAM использует для управления интерфейсом SDRAM. Ниже представлен обзор различных команд, используемых встроенным контроллером:

- **MRS**  
(установка регистра режима). Инициализирует рабочие параметры SDRAM во время последовательности включения.
- **PRE**  
(подзаряд). Осуществляет подзаряд банка.
- **ACT**  
(активация банка). Активирует страницу в запрашиваемом банке.
- **Read**  
Чтение из SDRAM.
- **Write**  
Запись в SDRAM.
- **REF**  
(регенерация). Переводит SDRAM в режим регенерации.
- **SREF**  
(саморегенерация). Помещает SDRAM в режим саморегенерации, в котором память управляет своими операциями регенерации изнутри.
- **NOP**  
(нет операций). Передается после чтения или записи с целью разрешения пакетных операций или с целью установления режима ожидания для различных доступов SDRAM. Во время этой команды MSSD3–0 деактивирован.

### **11.1.2.1 Команда установки регистра режима (MRS)**

Установка регистра режима является частью последовательности инициализации SDRAM. Каждая микросхема SDRAM имеет внутри регистр режима (количество регистров режима может достигать четырех) и команда MRS позволяет передать данные, используя разряды адреса ADDR13–0, в микросхему для записи в регистр. Таким образом, MRS инициализирует рабочие параметры SDRAM.

Регистр режима (регистр номер 0) имеет разрядность 13 бит и команда MRS инициализирует следующие параметры:

- Биты 2:0 – Burst length. Значение 0b111, что соответствует полной странице.
- Бит 3 – Burst type. Значение 0, что соответствует последовательному изменению адреса.
- Биты 6:4 – CAS Latency. Значение параметра определяется в соответствии с программированием регистра SDRCON[2:1].

Все оставшиеся биты регистра режима программируются нулевым значением.

При выполнении команды MRS, SDRAM должна находиться в состоянии подзаряда во всех своих банках.

Передача команды осуществляется соответствующей комбинацией значений на линиях управления. Эти значения приведены ниже (Таблица 63). Происходит запись сразу во все микросхемы памяти.

**Таблица 63 – Состояние выводов во время выполнения команды MRS**

<b>Разряд</b>	<b>Состояние</b>
MSSD3–0	0 (все)
CAS	0
RAS	0
SDWE	0
SDCKE	1
A14 == A30	0
A15 == A31	0 – MRS 1 – EMRS

Если в микросхеме памяти имеется дополнительный регистр режима и для инициализации требуется его запись, в регистре SDRCON должны быть сделаны соответствующие установки (бит EMREN должен быть равен 1). Во время последовательности инициализации контроллер SDRAM запишет нулевое значение в дополнительный регистр режима. Выполняется это с помощью той же команды MRS, но бит 15 адреса (и бит 31) будет установлен в 1, что соответствует дополнительному регистру режима. С помощью адресных линий 14 и 15 можно задать номер дополнительного регистра режима как 01 или 10.

### **11.1.2.2 Команда подзаряда (PRE)**

Особенность работы микросхемы памяти SDRAM состоит в том, что в некоторых ситуациях необходимо выполнять подзаряд (предварительный заряд) внутренних шин банков памяти. Команда PRE (Таблица 64) исполняется в двух ситуациях:

- Прерывание цикла чтения или записи. Точный момент, когда данная команда генерируется контроллером SDRAM, зависит от последовательности выполняемых транзакций и адресов, по которым выполняется доступ.

- Предварительная зарядка активного банка. При выполнении команды на линии A10 всегда высокий уровень, что соответствует подзаряду всех банков микросхемы. Однако подзаряжается только активный банк. Алгоритм работы контроллера SDRAM такой, что он поддерживает активным только один банк.

Передача команды PRE возможна в следующих случаях:

- Во время последовательной инициализации SDRAM.
- Перед командой АСТ (доступ к новой странице). Доступ к новой странице требует, чтобы активная страница была закрыта. Это делается путем выполнения команды подзаряда.
- Перед циклом регенерации. Выполнение команды регенерации требует, чтобы все банки памяти были подзаряжены.
- Перед тем, как процессор освобождает внешнюю шину. Это гарантирует однозначные условия начала работы нового мастера шины.

Передача команды PRE в микросхему памяти требует, чтобы в течение определенного времени к микросхеме не посылались команды активации. Данное время программируется в регистре SDRCON в соответствии с характеристиками микросхемы памяти.

**Таблица 64 – Состояние выводов во время выполнения команды PRE**

<b>Вывод</b>	<b>Состояние</b>
MSSD3–0	0 (все)
SDWE	0
RAS	0
CAS	1
SDCKE	1
SDA10	1

### **11.1.2.3 Команда выбора активного банка (АСТ)**

Команда АСТ (Таблица 65) посылается контроллером перед любым чтением или записью страницы, которая в данный момент не является активной. Перед командой АСТ идет команда PRE, если в каком-то банке имеется другая активная страница. Команда АСТ открывает доступ к странице SDRAM в некотором банке, и данная страница остается открытой до тех пор, пока не будет закрыта следующей командой предварительной зарядки PRE.

**Таблица 65 – Состояние вывода во время выполнения команды АСТ**

<b>Вывод</b>	<b>Состояние</b>
MSSD3–0	0 (один из четырех)
CAS	1
RAS	0
SDWE	1
SDCKE	1

### **11.1.2.4 Команда чтения (Read)**

Команда Read (Таблица 66) выполняет чтение данных из активной страницы памяти SDRAM. Если чтение страницы выполняется первый после активации

командой АСТ, между командами АСТ и Read должна быть задержка, которая определяется параметром tRCD. Если это уже не первое чтение, отслеживание данной задержки не выполняется. Команда Read задает начальный адрес слова в странице, с которого начинается чтение. Микросхема памяти будет использовать переданный адрес как стартовый и каждый такт увеличивать его, если необходимо выполнить последовательное чтение блока памяти. После подачи команды Read данные появляются на внешних выводах микросхемы через время равное задержке CAS Latency.

Одна транзакция чтения занимает различное количество циклов шины в зависимости от размера операнда и ширины внешней шины:

- чтение одинарного слова на 32-разрядной шине – 1 цикл;
- чтение одинарного слова на 16-разрядной шине – 2 цикла;
- чтение двойного слова на 32-разрядной шине – 2 цикла;
- чтение двойного слова на 16-разрядной шине – 4 цикла;
- чтение счетверенного слова на 32-разрядной шине – 4 цикла;
- чтение счетверенного слова на 16-разрядной шине – 8 циклов.

Если после команды Read на SDRAM не поступает какая-то другая команда, SDRAM продолжает последовательное чтение данных, наращивая внутренний адрес. Этот режим чтения называется «страничный режим» или «пакет». Такой режим удобен, когда транзакция чтения длится более одного цикла.

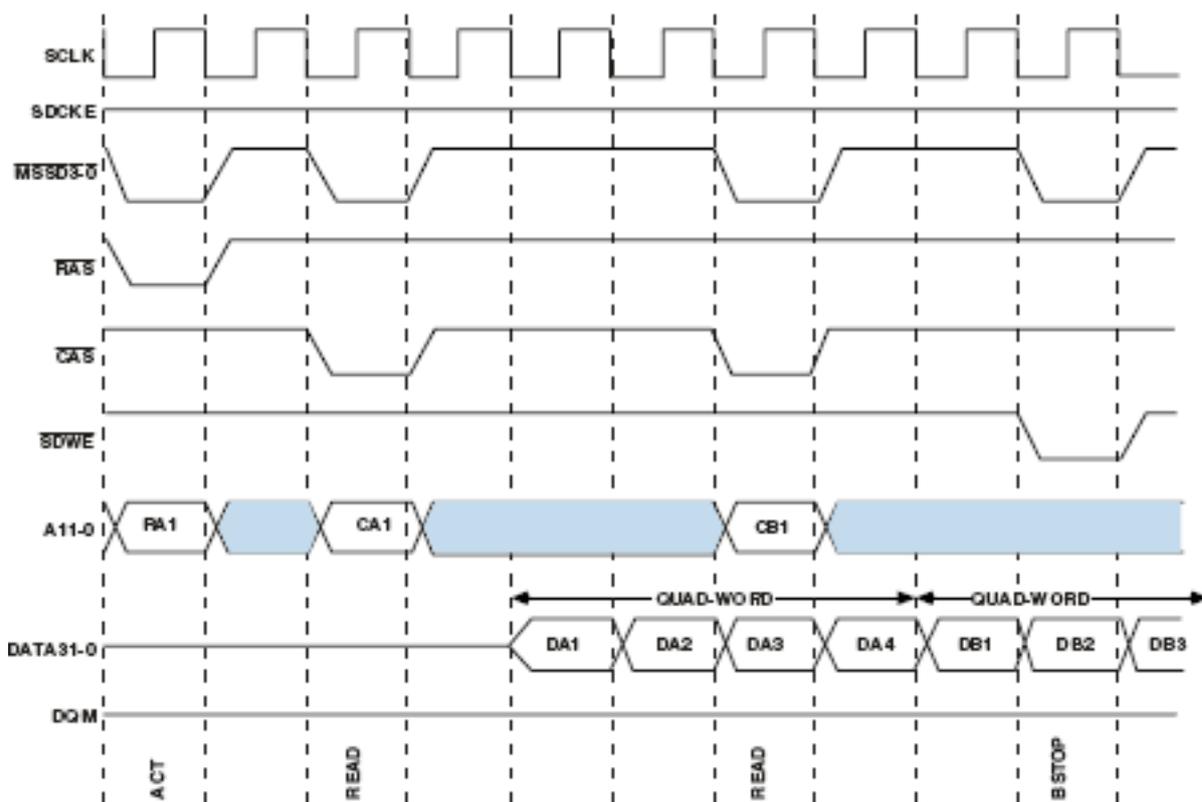
Когда транзакция полностью завершена, SDRAM может продолжить работу по нескольким путям:

- если есть следующая транзакция чтения SDRAM на той же странице, новая транзакция начинается с подачи команды Read;
- если нет новых транзакций SDRAM, контроллер посылает команду BSTOP. Эта команда прерывает последовательное чтение страницы;
- если после чтения приходит транзакция записи SDRAM на той же странице, команда BSTOP также останавливает чтение и запись начинается после приема данных;
- если есть доступ SDRAM к другой странице, поступает команда BSTOP и после нее посылается команда закрытия активной страницы PRE.

**Таблица 66 – Состояние вывода во время выполнения команды Чтение**

<b>Вывод</b>	<b>Состояние</b>
MSSD3–0	0 (один из четырех)
CAS	0
RAS	1
SDWE	1
SDCKE	1

Ниже приведена временная диаграмма использования команды Read для случая ширины шины 32 бита (Рисунок 29). Задержка CAS Latency равна двум.



**Рисунок 29 – Шина 32 разряда  
(пакетное чтение с последующим пакетным чтением на той же странице)**

#### 11.1.2.5 Команда записи (write)

Команда Write (Таблица 67) выполняет запись данных в активную страницу. Если это первая запись после активизации страницы, перед подачей команды записи должен быть соблюден интервал времени  $t_{RC D}$ . Команда записи устанавливает начальный адрес слова страницы, с которого начнется запись. Как и в случае чтения, если после записи нет других команд, на следующем такте будет выполняться запись по последовательно увеличенному адресу. Если запись нужно остановить, контроллер SDRAM должен передать команду B STOP.

Одна транзакция записи занимает разное количество циклов в зависимости от размера операнда транзакции и ширины внешней шины:

- запись одного слова на 32-разрядной шине – 1 цикл;
- запись одного слова на 16-разрядной шине – 2 цикла;
- запись двойного слова на 16-разрядной шине – 4 цикла;
- запись двойного слова на 32-разрядной шине – 2 цикла;
- запись счетверенного слова на 16-разрядной шине – 8 циклов;
- запись счетверенного слова на 32-разрядной шине – 4 цикла.

Одновременно с подачей начального адреса, процессор выставляет на шину данных и записываемые данные. Последовательная запись блока данных называется «страничный режим» или «пакет». При этом в последующих циклах нет необходимости подавать адрес, достаточно только передавать следующие данные.

Когда транзакция записи завершена, контроллер SDRAM может продолжить работу по нескольким путям:

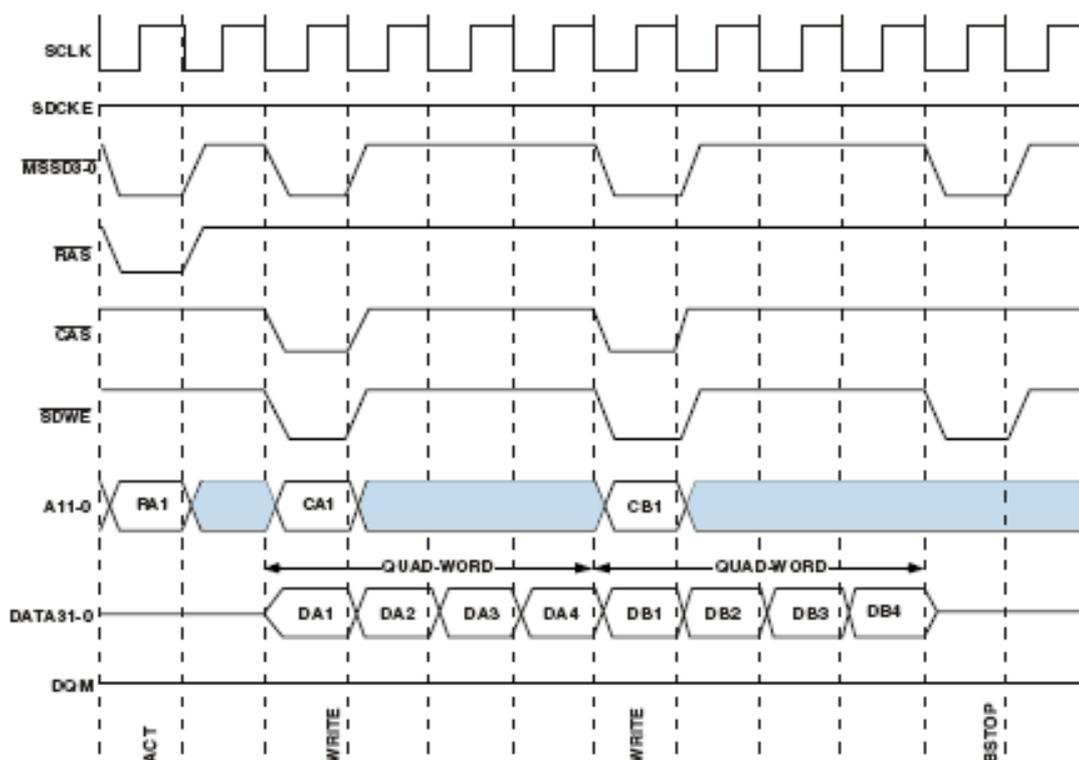
- если нет новых транзакций SDRAM, поступает команда B STOP, которая указывает на завершение записи блока данных;

- если следующая транзакция есть тоже запись в эту же страницу, контроллер может подавать новую команду записи с новым начальным адресом и данными;
- если следующая транзакция это чтение SDRAM на той же странице, команда BSTOP останавливает транзакцию записи и транзакция чтения начинается с нового цикла;
- если есть доступ SDRAM к другой странице, поступает команда BSTOP и далее подаются команды закрыть текущую страницу (PRE) и активизировать следующую. При этом соблюдаются все необходимые задержки, запрограммированные в регистре конфигурации.

**Таблица 67 – Состояние вывода во время выполнения команды Запись**

<b>Вывод</b>	<b>Состояние</b>
MSSD3-0	0 (один из четырех)
CAS	0
RAS	1
SDWE	0
SDCKE	1

Ниже приведена временная диаграмма использования команды Write для случая ширины шины 32 бита (Рисунок 30). Задержка CAS Latency равна двум.



**Рисунок 30 – Шина 32 разряда  
(пакетная запись с последующей пакетной записью на той же странице)**

### **11.1.2.6 Команда регенерации (REF)**

Память SDRAM требует периодической регенерации своих ячеек (строк) для сохранности информации. Команда REF (Таблица 68) является запросом к SDRAM

для выполнения транзакции регенерации. Эта команда генерируется автоматически процессором и вызывает регенерацию данных по адресу, который находится внутри SDRAM. Перед выполнением команды REF контроллер SDRAM выполняет проверку наличия активных страниц и если они имеются – выполняет команду предварительной зарядки (PRE) для активного банка.

Следующая команда активации (ACT) выдается контроллером только после минимальной задержки, равной  $t_{RC}$ , где:

$$t_{RC} = t_{RP} + t_{RAS} + 1$$

Частота регенерации (период подачи команд регенерации) устанавливается в регистре SDRCON.

**Таблица 68 – Состояние выводов во время выполнения команды REF**

<b>Вывод</b>	<b>Состояние</b>
MSSD3–0	0(все)
CAS	0
RAS	0
SDWE	1
SDCKE	1

#### **11.1.2.7 Команда саморегенерации (SREF)**

Описанная выше команда регенерации REF требует от процессора непрерывно отслеживать период повторения циклов регенерации и своевременно посылать команды REF. Эта команда может выполняться на фоне выполнения транзакций процессором к другим типам памяти (не SDRAM). Также существует режим, когда сама память отвечает за процедуру регенерации и самостоятельно поддерживает данные достоверными. Этот режим называется режимом саморегенерации и процессор переводит SDRAM в этот режим путем подачи команды SREF (Таблица 69). В данном режиме память недоступна для использования.

При выполнении команды саморегенерации должны быть выполнены все те же временные требования, что и при выполнении команды регенерации.

Перед выполнением команды SREF контроллер подзаряжает все банки микросхем. После команды SREF память выполняет операции регенерации самостоятельно без внешнего контроля. После выхода из режима саморегенерации контроллер SDRAM ждет количество циклов  $t_{RC}$  перед выполнением команды активизации банка (ACT).

Перевод памяти в режим саморегенерации осуществляется посредством установки 0-го бита регистра SDRSRF в единичное значение. Данный режим может быть использован как составная часть режима пониженного энергопотребления системы. Для выхода из режима саморегенерации необходимо очистить нулевой бит регистра SDRSRF.

**Таблица 69 – Состояние вывода во время выполнения команды SREF**

<b>Вывод</b>	<b>Состояние</b>
MSSD3–0	0 (все)
CAS	0
RAS	0
SDWE	1
SDCKE	0

Режим саморегенерации поддерживается тем, что на линии SDCKE постоянно низкий уровень, что запрещает использование входа синхронизации.

### **11.1.3 Вспомогательные регистры интерфейса внешней шины**

В процессоре 1967ВЦ3Т функции данных регистров ограничены.

#### **11.1.3.1 Регистр управления блокировкой шины (BUSLOCK)**

Регистр BUSLOCK определяет состояние запроса блокировки шины. В этом 32-разрядном регистре определен только один нулевой бит. Все другие биты (с 1 по 31) не используются. Запись значения 1 в нулевой бит вызывает запрос на блокировку шины со стороны процессора. В процессоре 1967ВЦ3Т внешний интерфейс всегда является мастером шины, и нет необходимости в регистре блокировки.

#### **11.1.3.2 Регистр статуса системы (SYSTAT)**

Регистр SYSTAT предназначен только для чтения и указывает на состояние некоторых параметров системы.

**Таблица 70 – Регистр SYSTAT**

<b>Бит</b>	<b>Имя</b>	<b>Назначение</b>
9:0	-	Всегда 0
10		Всегда 1
11	-	Всегда 0
12	BOOTMDE	Режим загрузки: 0 – из внешнего EPROM 1 – из другого источника
13	MRSCOMP	Признак окончания процедуры инициализации динамической памяти: 1 – завершена. Память готова к использованию. 0 – нет
14	BUSLKAСТ	Признак захвата шины: 1 – текущий процессор стал мастером на шине и удерживает ее. 0 – нет захвата шины
15	-	Всегда 0.
16	BROADREADERR	Ошибка доступа к адресному пространству 0x0С.....: 1 – была попытка чтения из указанного пространства 0 – нет ошибки
17	AUTODMAERR	Признак ошибки в работе каналов autoDMA: 1 – была попытка записи в выключенный канал 0 – нет ошибки
18	SDRAMERR	Признак ошибки доступа к СДРАМ. 1 – была попытка доступа неинициализированной внешней динамической памяти. 0 – нет ошибки
19	MPREADERR	Признак ошибки чтения процессором запрещенного адресного пространства: 1 – ошибка. 0 – нет ошибки
31:20	-	Всегда 0

Программы могут считывать регистр SYSTAT, используя имена SYSTAT или SYSTATCL(сброс SYSTAT):

- SYSTAT – чтение без изменений в содержимом регистра;
- SYSTATCL – разряды 19–16 будут сброшены после чтения.

### **11.1.3.3 Регистр ограничения времени управления шиной (BMAX)**

Регистр BMAX является 16-разрядным регистром общего назначения.

## **11.2 Поддержка мультипроцессорности**

Внешний порт не имеет возможностей для создания мультипроцессорных конфигураций с использованием общей параллельной шины.

## **11.3 Хост-интерфейс**

Процессор имеет возможность подключения внешнего хост-устройства через последовательный интерфейс. Используя 3-х проводной синхронный последовательный интерфейс можно выполнить доступ ко всем ресурсам процессора.

Процессор содержит синхронный последовательный интерфейс, который может быть использован для загрузки информации в процессор внешним ведущим устройством, а также для анализа состояния процессора посредством чтения его ресурсов. Интерфейс реализован на базе канала 13 контроллера DMA.

Интерфейс подключен к периферийной шине и, как и контроллер DMA, имеет базовый адрес **0x8000\_0000**.

### **11.3.1 Регистры интерфейса**

Краткое описание регистров интерфейса приведено ниже (Таблица 71).

**Таблица 71 – Регистры Хост-интерфейса**

<b>Номер\ смещение</b>	<b>Обозначение регистра</b>	<b>R/W</b>	<b>Описание</b>	<b>Состояние по сбросу</b>
0x005C	SDI	R	Адрес источника\ приемника	0
0x005D	CNT	R	Количество обменов	0
0x005E	MCR	R	Управление запросом прерываний к процессору	0
0x005F	HCR	R	Управление обменом	0
0x03E4	AUTODMA0	W	Буфер, разряды данных 31–0	-
0x03E5	AUTODMA1	W	Буфер, разряды данных 63–32	-
0x03E6	AUTODMA2	W	Буфер, разряды данных 95–64	-
0x03E7	AUTODMA3	W	Буфер, разряды данных 127–96	-

Хост-интерфейс использует два регистра канала 13 контроллера DMA. Регистр TCB доступен только по чтению и отражает информацию переданную ведущим устройством в интерфейс. Запись в регистр TCB может выполнять только хост-устройство. Регистр autoDMA канала 13 доступен только по записи и используется хост-устройством для чтения ресурсов процессора (внутренней и внешней памяти, периферийных устройств). Хост-устройство осуществляет программирование канала 13 контроллера DMA, после чего канал выполняет

поставленную задачу. Отметим, что канал 13 имеет наивысший приоритет среди каналов контроллера DMA. Один и тот же регистр TCB канала 13 может использоваться как для чтения, так и для записи данных.

### **11.3.2 Аппаратная реализация интерфейса**

Для организации обмена синхронный интерфейс использует три линии: линию синхронизации (клока) HCLK и линии данных HDI, HDO. Эти линии функционально совмещены с линиями задания конфигурации BOOT[2:0].

Во время сброса системы линии BOOT выполняют функцию задания варианта начального старта системы, а после завершения сброса они могут быть использованы для управления интерфейсом. Для реализации интерфейса необходимо использовать шесть линий:

- питание;
- общий;
- сброс nRST;
- BOOT.2/HCLK;
- BOOT.1/HDI;
- BOOT.0/HDO.

Начальная последовательность действий ведущего при этом будет следующей:

- установить на входе nRST низкий уровень (сброс системы) и задать на входах BOOT необходимую начальную конфигурацию;
- после некоторой задержки, установить на nRST высокий уровень, сохраняя на входах BOOT старое значение (обеспечивая время удержания);
- передать контроль над выводами BOOT ведущему устройству.

#### **11.3.2.1 Протокол обмена по последовательному интерфейсу**

- информация принимается каналом по положительному фронту (переход из 0 в 1) синхросигнала HCLK. Выдача информации осуществляется также по положительному фронту.
- Обмен всегда осуществляется 32-разрядными словами. Одна посылка состоит, как минимум из 34 бит: старт бит (0), 32 бита данных (старший бит первый) и стоп-бит (значение 1). Пример передачи данных приведен ниже (Рисунок 31).
- После сброса процессора хост-интерфейс некоторое время недоступен. Ведущее устройство после сброса процессора должно посылать сигнал синхронизации и значение 1 на входе данных HDI. Как только интерфейс включается (на него перестает действовать внутренний сброс процессора) на выходной линии данных появится активное значение 0 и через несколько тактов синхронизации – значение 1. После этого интерфейс готов к работе.
- Посредством интерфейса возможно выполнение операций чтения и записи, а также внутренних операций.



**Рисунок 31 – Прием информации контроллером интерфейса**

### **11.3.2.2 Стандартный формат обмена**

Имеется два формата команд: стандартный и короткий. В стандартном формате начальный адрес обмена передается отдельным словом, а в коротком он упакован в управляющее слово. Для выполнения операции записи в стандартном формате выполняется следующая процедура обмена:

- 1 Передается управляющее слово;
- 2 Передается начальный адрес;
- 3 Передаются данные.

Структура управляющего слова приведена ниже (Таблица 72).

**Таблица 72 – Структура управляющего слова стандартного формата**

<b>Номер</b>	<b>Название</b>	<b>Описание</b>
31:28	TYPE	Задают тип операции 00x0 – чтение 01x0 – запись 10xx – внутренняя операция 11xx – резерв
27:20	DDP	Эти биты загружаются в разряды DDP[31:24] регистра управления каналом 13
19:17	-	
16	MDF	1 – разрешение модификации адреса после каждого обмена
15:0	CNT	Количество слов данных при обмене. Загружается в регистр DDX

Для выполнения операции чтения стандартного формата выполняется следующая процедура обмена:

- 1 Передается управляющее слово;
- 2 Передается начальный адрес;
- 3 Принимаются данные.

При выполнении внутренней операции имеется возможность загрузить разряды 29:0 управляющего слова в специальный регистр управления интерфейсом MCR. В настоящее время используется только бит 0 регистра управления. При записи в него значения 1 происходит генерация запроса прерывания к процессору, т.е. имеется возможность послать запрос на прерывания, не выполняя процедуры обмена данными.

### **11.3.2.3 Короткий формат обмена**

Для выполнения операции записи в коротком формате выполняется следующая процедура обмена:

- 1 Передается управляющее слово;
- 2 Передается слово данных.

Структура управляющего слова приведена ниже (Таблица 73).

**Таблица 73 – Структура управляющего слова короткого формата**

Номер	Название	Описание
31:28	TYPE	Задают тип операции: 00x1 – чтение 01x1 – запись 10xx – внутренняя операция 11xx – резерв
27:20	DDP	Эти биты загружаются в разряды DDP[31:24] регистра управления каналом 13
19	A[31]	Бит 31 адреса. Используется для доступа к регистрам периферии
18:16	A[20:18]	биты адреса 20:18 при обращении к внутренней памяти. Все другие биты равны нулю
15:0	A[15:0]	Младшие биты адреса

При использовании короткого формата возможна запись или чтение только одного слова (в соответствии с размером) данных. Адрес не изменяется. Имеется возможность доступа ко всем ресурсам процессора (с ограничениями при доступе к внешней памяти).

Для выполнения операции чтения в коротком формате выполняется следующая процедура обмена:

- 1 Передается управляющее слово;
- 2 Принимаются данные.

Выполнение операции обмена хост-устройства с процессором сводится к программированию канала 13 контроллера DMA. Канал может выполнять как операцию записи (данные предоставляет хост-интерфейс), так и операцию чтения (при чтении данные загружаются по адресу AUTODMA и дальше по последовательному интерфейсу передаются в ведущее устройство). Имеется возможность доступа ко всем ресурсам процессора кроме чтения регистров ядра.

Управляющее слово после приема хост-интерфейсом загружается в регистр HCR и анализируется. Если формат обмена стандартный, принимается следующее слово, которое загружается в регистр SDI как адрес источника или приемника. Если формат обмена короткий, то загружаемое в SDI значение формируется как {HCR[19], 10'b0, HCR[18:16], 2'b00, HCR[15:0]}. Одновременно с загрузкой адреса, в счетчик обмена CNT загружается размер блока данных, который будет передан (прочитан или записан). В случае длинного формата значение счетчика будет равно HCR[15:0]. Для короткого формата загружаемое значение будет определяться длиной операнда, задаваемого битами HCR[22:21], т.е. это могут быть значения 1, 2 или 4. В регистре TCB канала КРДП имеется управляющее слово. Одновременно с заданием начального адреса и счетчика выполняется и загрузка бит управляющего слова. Используются только биты с 31 по 24 управляющего слова. В них загружается значение HCR[27:20]. Таким образом, можно задавать тип обмена, приоритет, длину слова. Разрешенные комбинации в разрядах DDP[31:29] являются:

- 010 – доступ к внутренней памяти;
- 011 – доступ к регистрам ядра;
- 100 – доступ к внешней памяти и регистрам периферийной шины;
- 110 – доступ к загрузочной флэш-памяти.

#### **11.3.2.4 Внутренняя операция интерфейса**

Внутренняя операция интерфейса задумывалась как средство управления ведущим устройством внутренних ресурсов интерфейса, а также для реализации различных взаимодействий с процессором. При выполнении внутренней операции содержимое принятого управляющего слова (биты с 0-го по 29-й) записываются в регистр MCR. Нулевой бит регистра MCR может формировать запрос прерывания к процессору. Данное прерывание анализируется в контроллере прерываний как запрос от канала 13 контроллера DMA. Процессор может прочитать значение регистра MCR.

#### **11.3.2.5 Вход синхронизации HCLK**

Вход синхронизации используется для задания тактового импульса обмена по последовательному интерфейсу. Прием и выдача информации осуществляется по положительному фронту HCLK. Отметим, что вход HCLK во время сброса выполняет функцию задания варианта начального старта(BOOT.2). Также этот вывод является входом-выходом порта общего назначения PC (бит 4). Во время сброса требуемый уровень на входе HCLK должен задаваться резистором (или внешним ведущим устройством). После сброса вход HCLK может управляться ведущим устройством (при его наличии) или иметь постоянное значение. Дело в том, что любое переключение на выводе PC[4] рассматривается интерфейсом как подача частоты HCLK. Поэтому данный вывод не может произвольно изменяться в случае отсутствия ведущего устройства.

#### **11.3.2.6 Вход данных HDI**

Вход используется для передачи процессору управляющих команд и данных. Во время сброса вход выполняет функцию задания варианта начального старта(BOOT.1). Также этот вывод является входом-выходом порта общего назначения PC (бит 5). Во время сброса требуемый уровень на входе HDI должен задаваться резистором (или внешним ведущим устройством). После сброса вход HDI может управляться ведущим устройством (при его наличии) или выполнять функции порта общего назначения PC[5]. При обмене с интерфейсом, ведущее устройство всегда посылает 32-разрядные данные с дополнительными старт и стоп битами. Если нет необходимости передавать информацию по входу HDI, на входе должен удерживаться высокий уровень. При наличии частоты синхронизации HCLK высокий уровень гарантирует отсутствие входных данных. Интерфейс обнаруживает наличие входных данных при принятии старт бита. После этого внутренний счетчик отсчитывает 32 такта данных. Стоп-бит нужен как дополнительный такт синхронизации для выполнения внутренних операций. Значение стоп-бита не контролируется. Однако для приема следующего слова, во входном потоке должна появиться 1 и только после нее будет детектирован старт бит.

#### **11.3.2.7 Выход данных HDO**

Выход данных HDO используется для передачи данных от процессора к ведущему устройству. В момент сброса процессора он выполняет функцию задания варианта начального старта, поэтому во время сброса данный вывод находится в отключенном состоянии. Если на входе HCLK имеются переключения, они рассматриваются как наличие частоты синхронизации от ведущего устройства. В этом случае данные с входа HDI записываются в приемный сдвиговый регистр. Если после завершения внутреннего сигнала сброса будет принято не менее 32-х единичных значений с входа HDI, интерфейс включится, и выход HDO станет

активным выходом. Для задания варианта начального старта на выводе HDO может использоваться резистор, подключенный к земле. Таким образом, на выводе HDO будет удерживаться уровень нуля до тех пор, пока интерфейс не включится и не начнет выдавать активную единицу. При чтении внутренней ячейки памяти процессора, внешнее хост-устройство заранее не знает, сколько тактов понадобится для считывания значения. Поэтому ведущее устройство после передачи команды чтения должно непрерывно подавать сигналы синхронизации HCLK и удерживать на входе HDI высокий уровень. Как только данные поступят в буфер интерфейса, на выводе HDO появится значение 0 (старт бит) и за ним будут переданы 32 бита данных. После них будет выдан стоп бит. Выдача данных на вывод HDO осуществляется автоматически, как только в буфере появляются данные. Поэтому имеется возможность послать данные ведущему устройству от процессора. Если в регистре HCR биты 31:30 не равны 01b (нет записи от ведущего устройства), процессор может выполнить запись в буфер канала 13 и передать информацию ведущему устройству. Отметим, что этот же буфер используется неявно ведущим устройством при записи информации в процессор. Буфер в этом случае выполняет функцию временного хранения информации. В режиме записи к нему нет доступа со стороны процессора.

### **11.3.2.8 Алгоритм работы внутренней машины состояний**

Внутренний автомат интерфейса тактируется клоком SOC-шины и постоянно анализирует поступающую информацию. Обработка информации осуществляется в соответствии со следующим алгоритмом:

Состояние S0 Ожидание управляющего слова. Если слово принято – запись слова в регистр HCR и переход в состояние S1.

Состояние S1 Анализ регистра HCR. Если стандартный формат чтения – переход в состояние S4. Короткий формат чтения – переход в состояние S7. Если стандартный формат записи – переход в состояние S8. Короткий формат записи – переход в состояние S10. Если внутренняя операция – переход в состояние S2. В случае резервной операции – зависание машины состояний.

Состояние S2 Выполнение внутренней операции. Перезапись регистра HCR в регистр MCR. Переход в исходное состояние S0.

Состояние S4 Ожидание адреса для выполнения чтения. После приема адреса – запись управляющей информации (начальный адрес, счетчик слов, управляющее слово) в TCB канала. Переход в состояние S6 – ожидания выполнения операции чтения. Если в момент записи в TCB канал активен и управляющее слово также активно – устанавливается флаг ошибки и канал выключается.

Состояние S6 Ожидание завершения работы канала. Канал завершит работу только после чтения всего блока данных. При этом моментом окончания работы канала является факт загрузки последнего слова в буфер и начало его передачи в ведущее устройство.

Состояние S7 Запись управляющей информации (начальный адрес, счетчик слов, управляющее слово) в TCB канала. Переход в состояние S6 – ожидания выполнения операции чтения. Если в момент записи в TCB канал активен и управляющее слово также активно – устанавливается флаг ошибки и канал выключается.

Состояние S8 Ожидание адреса для выполнения операции записи. После приема адреса – запись управляющей информации (начальный адрес, счетчик слов, управляющее слово) в TCB канала. Переход в состояние S9 – ожидания приема данных для записи. Если в момент записи в TCB, канал активен и управляющее слово также активно – устанавливается флаг ошибки и канал выключается.

**Состояние S9** Ожидание данных для выполнения операции записи. После приема данных – запись принятого слова в буфер канала. Ожидание нового слова данных или завершения работы канала. Канал инициирует запись данных в процессор, только если в буфере данных достаточно данных для запрограммированной записи.

**Состояние S10** Запись управляющей информации (начальный адрес, счетчик слов, управляющее слово) в TCB канала. Переход в состояние S9 – ожидания данных для выполнения операции записи. Если в момент записи в TCB канал активен и управляющее слово также активно – устанавливается флаг ошибки и канал выключается.

Машина состояний интерфейса требует корректного завершения всех операций. Нет возможности ведущему процессору анализировать состояние интерфейса, и в случае различных аномалий возвращать его функционирование в начальное состояние. Только сброс всего процессора является выходом из возможной ошибочной ситуации.

### **11.3.2.9 Доступ к регистрам ядра**

Выше было отмечено, что в случае, когда биты DDP[31:29] управляющего регистра равны 011'b, хост устройство может осуществлять доступ к внутренним регистрам вычислительного ядра процессора. Отметим, что это единственный способ получить возможность доступа к регистрам для внешнего устройства. Для доступа к регистрам используются только разряды адреса с 0-го по 10-й. Биты с 5-го по 10-й образуют номер группы регистров, а биты с 0-го по 4-й номер регистра в группе. Номера групп при доступе со стороны хост-устройства совпадают с номерами групп, которые используются системой команд процессора для выполнения пересылок. Используемые номера групп приведены ниже (Таблица 73).

**Таблица 74 – Группы регистров вычислительного ядра**

<b>Номер группы</b>	<b>Название группы</b>
0	Регистры общего назначения вычислительного модуля X: XR[0-31]
2	Регистры общего назначения вычислительного модуля Y: YR[0-31]
10	Регистры модуля отладки
12	Регистры общего назначения модуля IALU : J[0-31]
13	Регистры общего назначения K модуля IALU : K[0-31]
14	Дополнительные регистры J модуля IALU : JB[0-3], JL[0-3]
15	Дополнительные регистры K модуля IALU : KB[0-3], KL[0-3]
26	Регистры устройства управления
27	Регистры модуля отладки
30	Регистры модуля защиты и управления кэш-памятью
31	Регистры модуля защиты и управления кэш-памятью

### **11.3.2.10 Ограничения интерфейса**

При режиме старта BOOT==000 процессор переходит в режим ожидания прерывания. Хост-устройство может выполнить загрузку данных в память процессора и запустить процессор на выполнение программы. Последовательный хост-интерфейс позволяет ведущему устройству осуществлять полный контроль процессора. Передача данных по последовательному интерфейсу выполняется с частотой HCLK, а все внутренние операции процессора, в контроллере интерфейса, с частотой SOC-шины. Чтобы не было проблем с переполнением приемного буфера интерфейса желательно иметь частоту HCLK не выше частоты SOC-шины. Особое

внимание необходимо уделить начальным операциям, выполняемым после сброса, т.к. в этом случае процессор работает на входной частоте синхронизации, и она может быть недостаточно высокой. В этом случае можно анализировать параметры частот процессора и менять частоту обмена HCLK. Особое внимание нужно уделять операциям записи блока данных в процессор, т.к. в этом случае внутренние операции должны успевать выполняться до приема новой порции данных.

Операции чтения и записи интерфейса независимы. Интерфейс всегда начинает выдачу данных на выход HDO, если в его буфер поступают данные. Таким образом, выдачу данных можно вести одновременно с приемом новой информации. Единственное ограничение это совместное использование буфера интерфейса, как для операций чтения, так и для операций записи. Поэтому операция записи может начинаться только после завершения чтения последнего слова данных.

Для корректной работы интерфейса рекомендуется не начинать новой операции до момента полного завершения предыдущей.

## 12 Порты связи

Процессор имеет два LVDS порта связи для обеспечения 8\4\1-битного приема и 8\4\1-битной передачи в мультипроцессорных системах.

Порты связи имеют следующие характеристики:

- тактовая скорость связи может быть выбрана из  $x1/4$ ,  $x1/2$ ,  $x2/3$ , или  $x1$  от внутренней частоты процессора;
- данные порта связи упакованы в 128-битные слова для DMA передачи во внутреннюю и внешнюю память;
- каждый порт связи имеет собственные буферные регистры;
- передачи каждого порта контролируются подтверждающим протоколом;
- порты связи поддерживают полный дуплекс и передачу в\из внешних портов или других каналов связи.

Порты связи предназначены для использования при двухточечной связи между процессорами в системе, но могут использоваться в качестве интерфейса для коммуникации с любыми другими устройствами, разработанными для работы по тому же протоколу.

Порты связи процессора используют схему LVDS (дифференциальные сигналы низкого напряжения). Данные выдаются и принимаются как на возрастающем, так и на спадающем фронте тактового сигнала. Каждый из портов имеет канал приема и передачи, и способен работать в полностью дуплексном режиме.

Управление передачей ядром процессора выполняется посредством механизмов прерывания и опроса. Управление передачей контроллером DMA выполняется через назначенные DMA-каналы приема/передачи. При этом все каналы DMA портов поддерживают цепочки операций, а также могут быть использованы при загрузке процессора во время старта.

### 12.1 Архитектура портов связи

Порты связи подключены к SOC-шине как периферийное устройство (Рисунок 32).

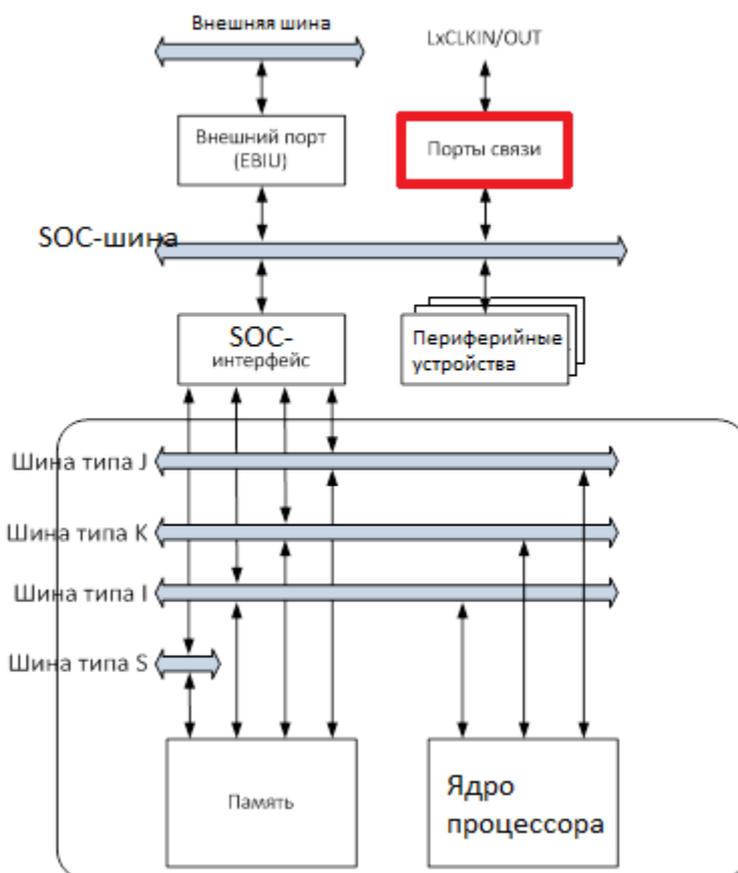


Рисунок 32 – Подключение портов связи на кристалле

Порт связи состоит из двух частей – передатчик и приемник. Каналы передачи и приема имеют буфер, как показано ниже (Рисунок 33). Буферные регистры, подключенные к SOC-шине, доступны программисту как регистры LBUFTX и LBUFRX. Они являются 128-разрядными, отображаемыми в памяти универсальными регистрами. Регистры дополнительных буферов и сдвиговые регистры являются программно недоступными.

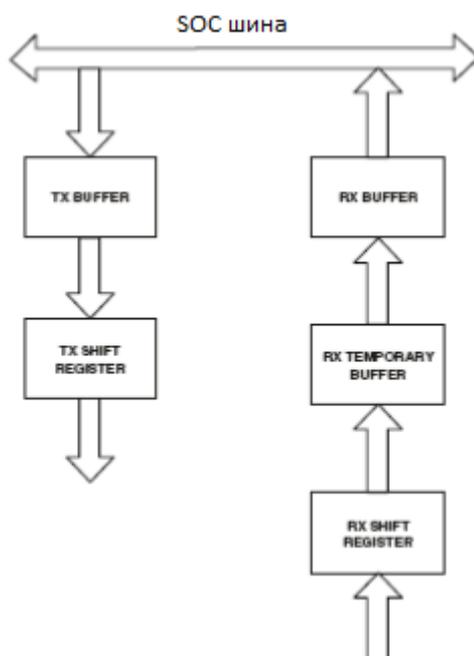


Рисунок 33 – Архитектура порта связи

Дополнительная буферизация позволяет конвейеризировать операции приема-передачи порта связи и операции контроллера ДМА. Также имеет значение, что сдвиговые регистры приема-передачи имеют собственную частоту работы, которая отличается от частоты SOC-шины.

## **12.2 Внешние выводы портов связи**

Для выполнения обмена данными с внешними устройствами порты связи используют внешние выводы микросхемы, которые описаны ниже (Таблица 75). В таблице приведен перечень выводов одного порта связи. Знак 'x' в имени сигнала указывает на порт связи – 0 или 1. В набор внешних выводов порта связи входят как дифференциальные выводы, так и обычные выводы. Для каждого дифференциального вывода существует пара линий: P и N. На линии P передается истинное значение сигнала, а на линии N противоположное. Раздельные шины данных для приема и для передачи имеют разрядность 8 бита каждая. При этом имеется режим передачи с использованием только одного разряда шины данных из восьми.

**Таблица 75 – Описание выводов – порты связи**

<b>Сигнал</b>	<b>Описание</b>
LxDAT07-0P	Шина данных 7-0 передача LVDS P
LxDAT07-0N	Шина данных 7-0 передача LVDS N
LxCLKOUTP	Тактовый генератор передачи LVDS P
LxCLKOUTN	Тактовый генератор передачи LVDS N
LxACKI	Входной сигнал подтверждения передатчика. Используя этот сигнал приемник указывает передатчику, что можно продолжать передачу
LxBCMPO	Завершение блока. Когда передача выполняется с использованием ДМА, данный сигнал указывает приемнику, что передаваемый блок закончился. Во время сброса выводы L1BCMPO, L2BCMPO и L3BCMPO используются как входы для конфигурирования процессора. Перед началом инициализации порта, порт связи выдает сигнал высокого уровня на LxBCMPO (деактивирован), указывая приемнику, что порт подсоединен.
LxDAT17-0P	Данные 7-0 прием LVDS P
LxDAT17-0N	Данные 7-0 прием LVDS N
LxCLKINP	Тактовый генератор приема LVDS P
LxCLKINN	Тактовый генератор приема LVDS N
LxACKO	Выходной сигнал подтверждения приемника. Используя этот сигнал приемник указывает передатчику, что можно продолжать передачу.
LxBCMPI	Завершение блока. Когда передача выполняется с использованием ДМА, данный сигнал указывает приемнику, что передаваемый блок закончился. После сброса сигнал высокого уровня на LxBCMPI указывает на то, что к приемнику порта подключен внешний передатчик.

Каждый порт связи имеет два независимых канала, один для приема и второй для передачи, которые могут работать одновременно. Канал передачи передает данные на другое устройство, а канал приема получает данные с другого устройства. Каждый канал осуществляет обмен данными используя до восьми бит

данных и используя сигналы LxCLKOUTP/N, LxACKI, LxCLKINP/N и LxACKO для управления передачей данных.

Сигналы LxVCMPI и LxVCMPO используются для информирования о том, что текущая передача блока данных завершена. Порты связи должны соединяться так, как это показано ниже ( ).

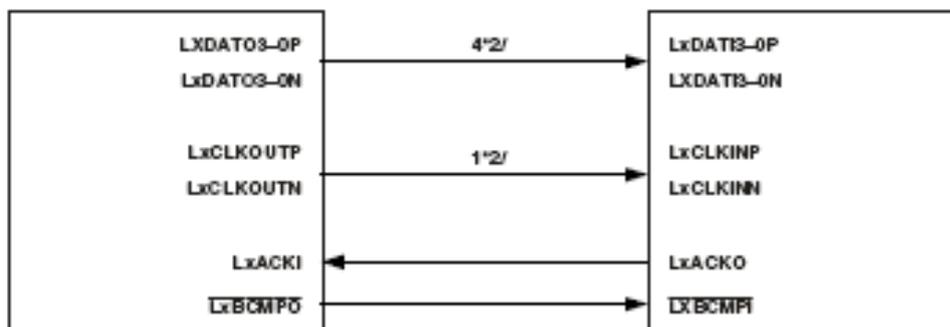


Рисунок 34 – Конфигурация порта связи (4-разрядный режим)

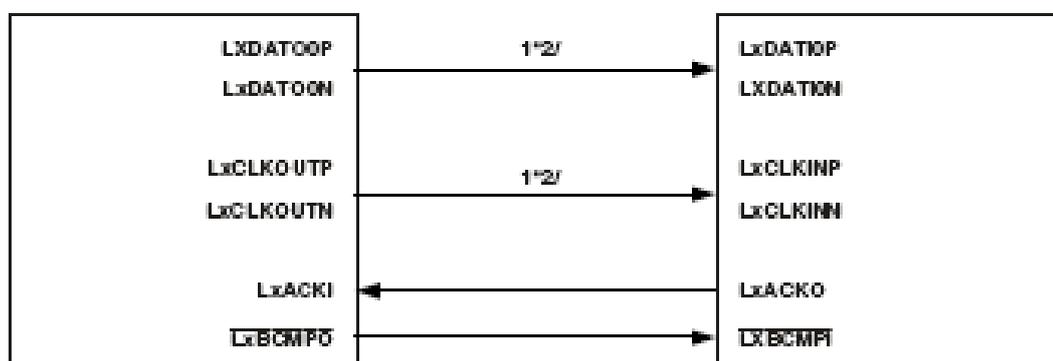


Рисунок 35 – Конфигурация порта связи (1-разрядный режим)

Порты связи процессора являются тактируемыми высокоскоростными LVDS портами данных. LVDS является стандартом для дифференциальной передачи сигналов между удаленными элементами. LVDS обеспечивает более высокую частоту, более высокий уровень устойчивости к шумам, более низкое энергопотребление и меньшее число электромагнитных помех.

Передача сигналов LVDS требует дифференциального завершения. Рисунок 36 показывает процесс передачи через порт от одного процессора к другому. Внутренние 100 Ом резисторы завершения (RT) на стороне приемника обеспечивают выполнение данного требования. Линии на плате должны быть по возможности одинаковыми для того, чтобы поддерживать одинаковое время задержки для всех выводов данных и тактовых сигналов.

Линии несимметричных сигналов (LxACKI, LxACKO, LxVCMPI и LxVCMPO) не так критичны, но их задержки должны были быть близки к задержкам соответствующих им дифференциальных сигналов.

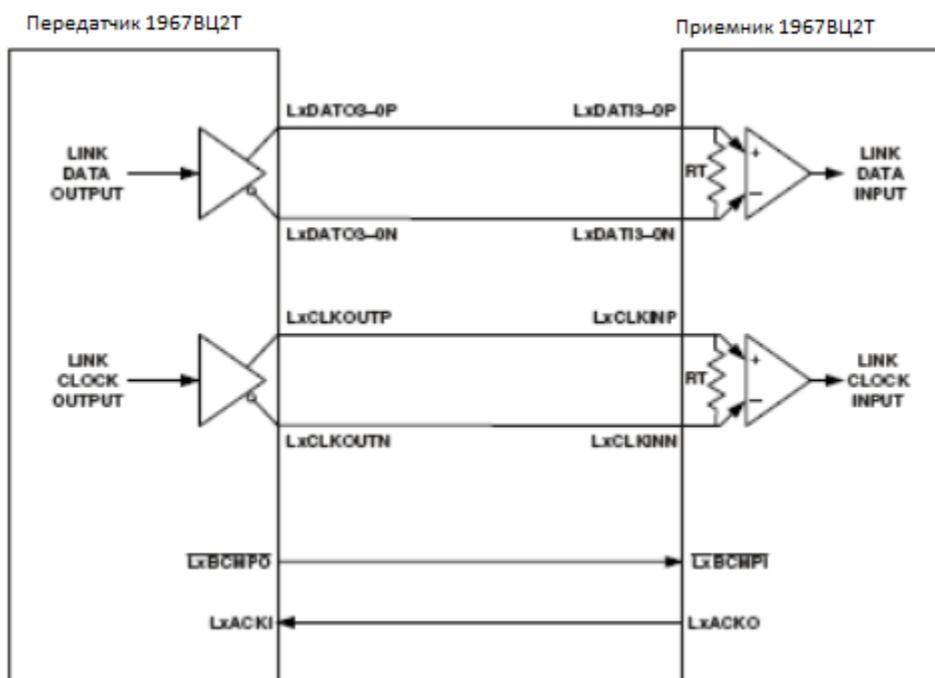


Рисунок 36 – Конфигурация передачи между процессорами

### 12.3 Группа регистров портов связи

Регистры буфера портов связи доступны как счетверенные слова. Две группы регистров данного типа описаны ниже (Таблица 76). Регистры управления и статуса доступны как одинарные слова.

Таблица 76 – Группа регистров буфера приема/передачи порта связи

Имя	Описание	Адрес	Значение по умолчанию
LBUFTX0:0	Данные передатчика порта связи 0	0x 04A0	Не определено
LBUFTX0:1	Данные передатчика порта связи 0	0x 04A1	Не определено
LBUFTX0:2	Данные передатчика порта связи 0	0x 04A2	Не определено
LBUFTX0:3	Данные передатчика порта связи 0	0x 04A3	Не определено
LBUFRX0:0	Данные приемника порта связи 0. только чтение	0x 04A4	Не определено
LBUFRX0:1	Данные приемника порта связи 0. только чтение	0x 04A5	Не определено
LBUFRX0:2	Данные приемника порта связи 0. только чтение	0x 04A6	Не определено
LBUFRX0:3	Данные приемника порта связи 0. только чтение	0x 04A7	Не определено
LBUFTX1:0	Данные передатчика порта связи 1	0x 04A8	Не определено
LBUFTX1:1	Данные передатчика порта связи 1	0x 04A9	Не определено
LBUFTX1:2	Данные передатчика порта связи 1	0x 04AA	Не определено
LBUFTX1:3	Данные передатчика порта связи 1	0x 04AB	Не определено
LBUFRX1:0	Данные приемника порта связи 1. только чтение	0x 04AC	Не определено
LBUFRX1:1	Данные приемника порта связи 1. только чтение	0x 04AD	Не определено

LBUFRX1:2	Данные приемника порта связи 1. только чтение	0x 04AE	Не определено
LBUFRX1:3	Данные приемника порта связи 1. только чтение	0x 04AF	Не определено

**Таблица 77 – Группа регистров статуса и контроля порта связи**

<b>Имя</b>	<b>Описание</b>	<b>Адрес</b>	<b>Значение по умолчанию</b>
LRCTL0	управление приемника порта связи 0	0x 00E0	0x0000 0001 или 0x0000 0011
LRCTL1	управление приемника порта связи 1	0x 00E1	0x0000 0001 или 0x0000 0011
LTCTL0	управление передатчика порта связи 0	0x 00E4	0x0000 0000
LTCTL1	управление передатчика порта связи 1	0x 00E5	0x0000 0000
-	-	0x 00E6– 0x 00EF	Нет данных
LRSTAT0	статус приемника порта связи 0	0x 00F0	0x0000 0000
LRSTAT1	статус приемника порта связи 1	0x 00F1	0x0000 0000
LTSTAT0	статус передатчика порта связи 0	0x 00F4	0x0000 0002
LTSTAT1	статус передатчика порта связи 1	0x 00F5	0x0000 0002
LRSTATC0	сброс статуса приемника порта связи 0	0x 00F8	Не определено
LRSTATC1	сброс статуса приемника порта связи 1	0x 00F9	Не определено
LTSTATC0	сброс статуса передатчика порта связи 0	0x 00FC	Не определено
LTSTATC1	сброс статуса передатчика порта связи 1	0x 00FD	Не определено

## 12.4 Прием и передача данных

Обмен данными осуществляется посредством записи в буфер передачи и чтением из буфера приема. Длина передаваемых данных всегда равна 128 битам. Все данные, записанные в буфер передачи, копируются в сдвиговый регистр, как только он становится пустым, и после этого передаются. Приемник разрешает продолжение приема данных, только когда его сдвиговый регистр пуст или когда в его буферных регистрах есть достаточно места для приема данных из сдвигового регистра, когда прием этого счетверенного слова будет завершен. После того, как целое счетверенное слово получено, приемник перемещает данные из сдвигового регистра в буфер приема, когда он свободен.

Если приемник не готов принять следующее слово данных, он использует линию подтверждения LxACK0 для приостановки работы передатчика.

В таблицах (Таблица 76, Таблица 77) приведены имена регистров, которые пользователь может использовать при программировании портов связи.

## 12.5 Связь с DMA

Каждый порт связи может соединяться с двумя каналами DMA. Один канал используется для передачи данных, в то время как второй используется для приема данных. Оба канала DMA связаны через интерфейс с внутренней памятью, внешней памятью или другими буферами портов связи. Передатчики портов связи с 0 по 1-й соединены соответственно с 4-го по 5-й каналами DMA. Приемники портов связи с 0 по 1-й соединены соответственно с 8-го по 9-й каналами DMA. Каналы DMA с 4-го по 5-й называются каналами передачи т.к. передают данные передатчикам порта связи,

а каналы с 8 по 9-й называются каналами приема т.к. принимают данные от приемников порта связи.

Передачик порта связи формирует сервисный запрос к каналу передачи DMA, когда буферный регистр LBUFTX пуст и канал DMA включен. Приемник порта связи формирует запрос к каналу приема DMA, когда он записывает счетверенное слово данных в буферный регистр LBUFRX и канал DMA включен.

Канал-приемник DMA (с 8 по 9) может также использоваться для транзитных передач посредством записи данных из приемника своего порта связи в буферный регистр передатчика любого другого порта связи. Канал DMA при этом отслеживает, что буферный регистр передатчика свободен.

## **12.6 Завершение блочной передачи**

Функция завершения передачи блока позволяет передатчику информировать приемник о том, что блочная передача завершена.

Поскольку канал передачи DMA точно знает, сколько слов необходимо передать и может отследить момент передачи последнего слова, то вместе с записью последнего слова в буфер передатчика, он формирует сигнал завершения передачи блока и передает его в передатчик вместе с данными. Порт связи активизирует LxVCMPO, когда он передает последнее счетверенное слово и в регистре управления LTCTL установлен разряд VCMPE.

Когда вход LxVCMPI активен, то принимающий порт связи передает эту информацию каналу DMA вместе с запросом. Это позволяет каналу приемнику DMA отследить завершение приема и закончить свою работу.

## **12.7 Прерывания портов связи**

Порты связи имеют специализированные прерывания для управления потоком данных, когда порт связи осуществляет передачу с использованием ядра процессора (в отличие от передачи с использованием каналов DMA). Прерывание от приемника порта связи является активным только тогда, когда канал DMA для порта связи не включается. Порт приема активизирует прерывание, когда счетверенное слово, получаемое портом связи, ожидает в регистре LBUFRX. Прерывание приема порта связи является чувствительным по уровню и поэтому, если канал DMA, взаимодействующий с портом связи, становится активным после разрешения прерывания порта связи, прерывание приема порта связи будет деактивировано и будет сформирован запрос к DMA.

## **12.8 Инициализация после сброса и загрузка через порт связи**

Старт процессора может быть выполнен посредством загрузки начального кода программы через порты связи. В этом режиме порт работает как подчиненное устройство, настроенное на прием данных. После сброса все каналы DMA порта приема могут настраиваться для передачи блока из 256 слов во внутреннюю память по адресам с 0 до 255 и настраиваются для формирования прерывания по окончании передачи блока. Соответствующие вектора прерываний для каналов DMA устанавливаются в нулевой адрес.

При сбросе или запрещении портов связи через регистры управления, содержимое буферов порта связи сбрасывается.

## 12.9 Протокол передачи данных порта связи

Порт связи осуществляет передачу данных через 1-, 4- или 8-разрядную шину данных по каждому направлению (TX или RX), с использованием трех типов контрольных сигналов (Таблица 75). В одноразрядном режиме для передачи используются выводы LxDATI0P/N и LxDATO0P/N.

Каждый порт связи имеет два независимых канала, которые могут работать одновременно. Канал передачи передает данные на внешнее устройство, канал приема получает данные с внешнего устройства. Каждый канал осуществляет передачу данных с использованием одного, четырех или восьми разрядов данных, используя сигналы LxCLKOUTP/N, LxACKI, LxCLKINP/N и LxACKO для управления передачей данных. Сигналы LxBCMPI и LxBCMPO используются для оповещения о завершении текущей передачи блока.

Существует несколько общих правил, которые применяются в протоколе порта связи (Рисунок 37 – Рисунок 44). Знание этих правил позволяет понять примеры, приведенные на диаграммах. Общие правила таковы:

- первые данные передаются на нарастающем (из 0 в 1) фронте тактового сигнала порта связи (условно LXCLKOUTP);
- последние данные передаются на спадающем (из 1 в 0) фронте тактового сигнала порта связи (условно LXCLKOUTP);
- LXCLKOUTP переводится в низкий уровень, когда порт связи находится в состоянии простоя.

Минимальный объем передачи данных через порт равен счетверенному слову. Счетверенное слово может быть передано:

- в течение 8 тактовых циклов, когда используется 8 разрядов шины данных;
- в течение 16 тактовых циклов, когда используются четыре разряда шины данных;
- в течение 64 циклов, когда используется только один разряд данных.

На рисунках ниже показаны случаи обмена для различных шин данных.

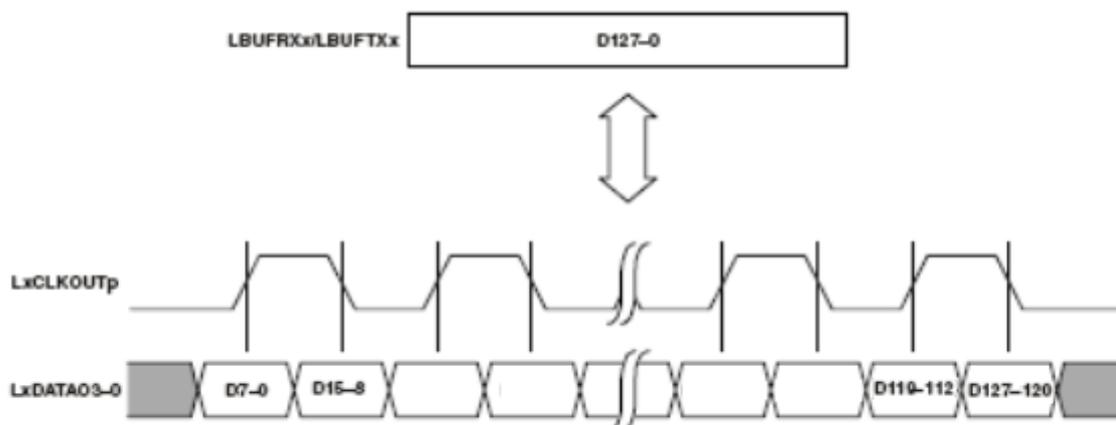


Рисунок 37 – Шина данных 8 разрядов

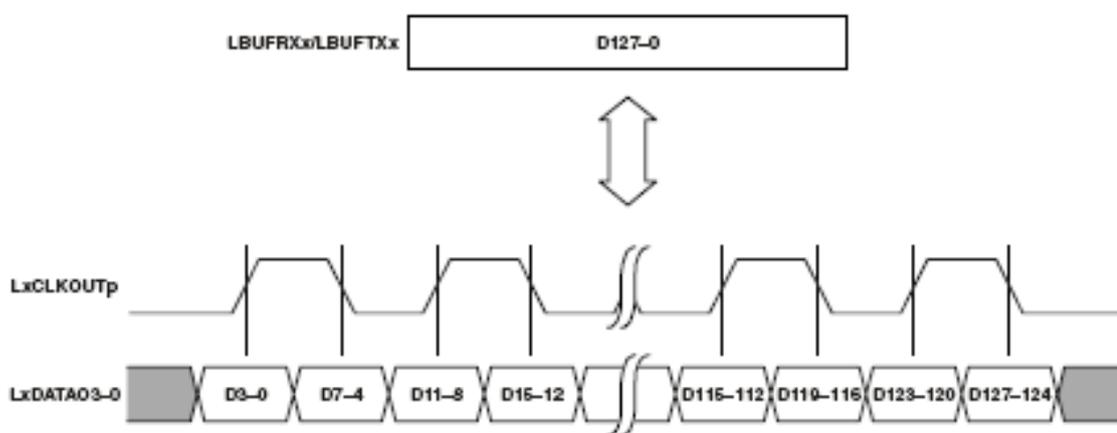


Рисунок 38 – Шина данных 4 разряд

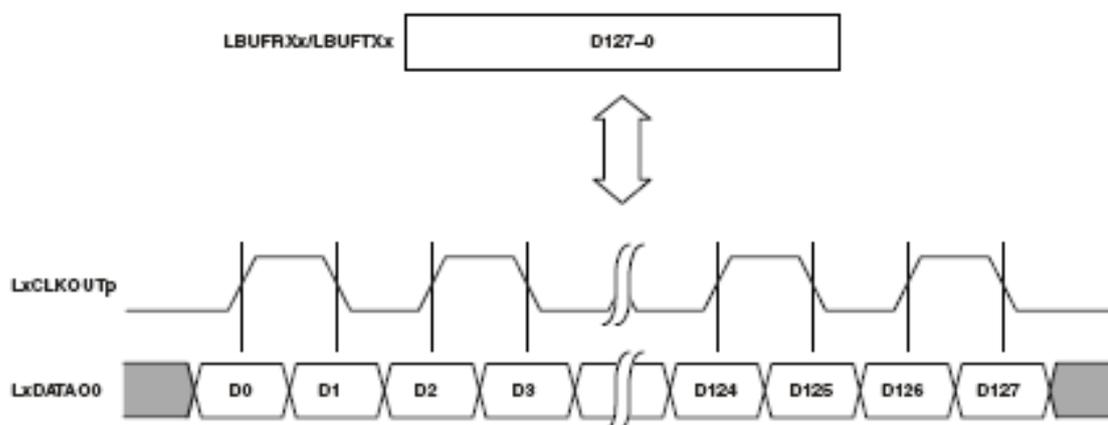


Рисунок 39 – Шина данных 1 разряд

Передача данных начинается, когда на LxACKI высокий уровень сигнала, что обозначает, что буфер приема свободен. Как показано на рисунках, первые биты данных достоверны до первого нарастающего фронта LxCLKOUTP и следующие за ними биты данных достоверны до спадающего фронта тактового сигнала.

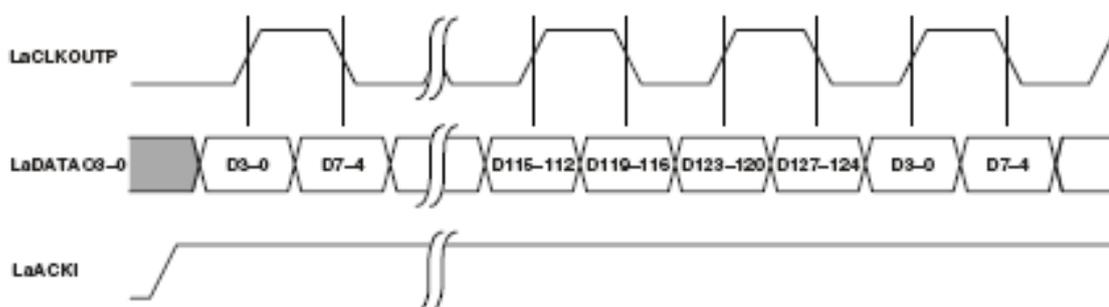


Рисунок 40 – Порт связи “а” передает слова порту связи “b” одно за другим

Рисунок 41 иллюстрирует состояние порта связи “b”, когда он не готов принимать больше данных. В этом случае после передачи текущего слова данных передатчик будет ждать пока сигнал подтверждения не примет высокий уровень.

На последовательность передачи влияет установка разрядов верификации данных. Это разряд RVERE в регистре LRCTL и разряд TVERE в регистре LTCTL. Если установлен разряд TVERE, байт контрольной суммы отправляется после последнего байта в счетверенном слове. За байтом контрольной суммы всегда следует «пустой» байт. Отправка байта контрольной суммы (биты Vx) и «пустого» байта (Xx) выполняется в течение двух циклов для шины 4-разрядных данных и восемь циклов для шины 1-разрядных данных. Рисунок 42 иллюстрирует последовательность передачи для случая 1-разрядной шины данных.

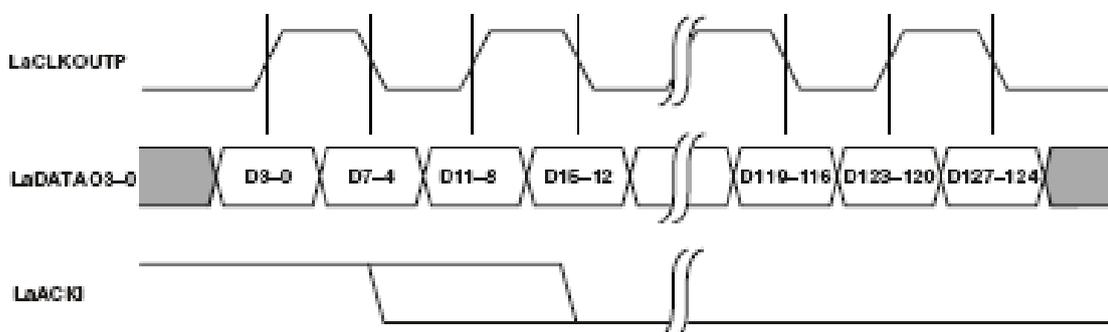


Рисунок 41 – Порт связи “а” передает порту связи “b” (приемник заполнен)

Сообщения между различными устройствами в системе обычно имеют различную длину. В некоторых случаях блоки данных также имеют различную длину. При этом знание длины блока не всегда доступно для принимающего устройства и не может быть передано передающим устройством. В этом случае на помощь приходят управляющие сигналы LxVCMPO и LxVCMPI.

Передающий порт связи указывает порту связи приема, что блок завершен с использованием выходного сигнала LxVCMPO передатчика, который связан с входным сигналом приемника LxVCMPI.

Когда приемник распознает этот сигнал, он передает информацию каналу DMA о том, что блок данных завершен. В результате этого канал DMA игнорирует то, что его счетчик слов еще не достиг значения нуля, и работает так, как при завершении блока.

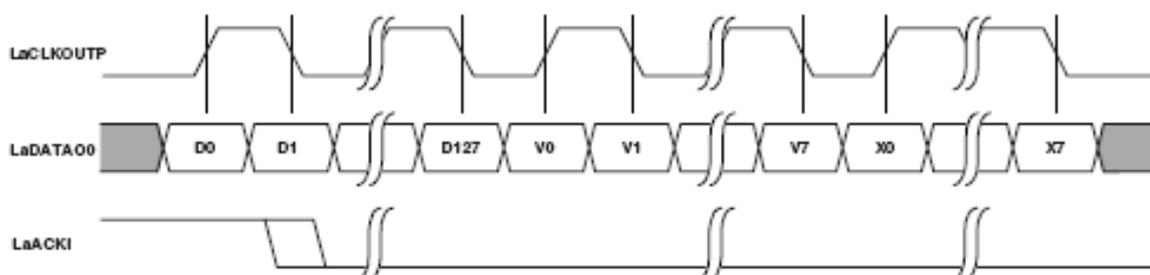


Рисунок 42 – Порт связи “а” передает порту связи “b” с верификацией

Сигнал LxVCMPO указывает на завершение блока переходом на низкий уровень после первого нарастающего фронта LxCLKOUT в последнем счетверенном слове блока. На LxVCMPO устанавливается высокий уровень сигнала (неактивный) перед последним спадающим фронтом сигнала LxCLKOUT того же счетверенного слова. Сигнал является неактивным, если разряд TVCMPE сбрасывается в регистр LTCTL или когда передачи порта связи выполняются под управлением ядра процессора.

Рисунок 43 показывает как порт связи “а” передает сигналу порту связи “b” о том, что текущий буфер завершен.



Рисунок 43 – Порт связи “а” передает порту связи “b” с завершением блока

Порт связи “а” передает сигналы порту связи “b” о том, что текущий буфер заполнен (Рисунок 44). Новое счетверенное слово следует за завершенным блоком.



Рисунок 44 – Порт связи “а” передает порту связи “b” с завершением блока (новый блок из порта связи “а” следует немедленно после завершения первого блока)

## 12.10 Задержки передачи через порт связи

Порты связи должны быть в состоянии преодолевать задержки между источником и пунктом назначения. Задержки для различных типов сигналов (LxCLKOUT и LxDAT7–O) должны соответствовать допускам, указанным в техническом описании. Существует два типа ограничений для задержек в системе:

- задержка от LxCLKOUT в передатчике к LxCLKIN в приемнике плюс задержка от LxACKO в приемнике к LxACKI в передатчике должны быть меньше чем половина периода передачи счетверенного слова.
- максимальная разница между задержкой трассировки LxVCMPO и задержкой трассировки тактового сигнала (и данных) составляет одну треть периода передачи счетверенного слова.

Первое ограничение связано со следующим фактором. Если передатчик отправляет данные, а приемник передает сигнал о том, что он не готов принять больше данных через LxACKO, передатчик должен сначала обработать LxACKI до

того, как он отправит следующие данные. Если LxACKI не проанализирован вовремя, данные будут утеряны приемником.

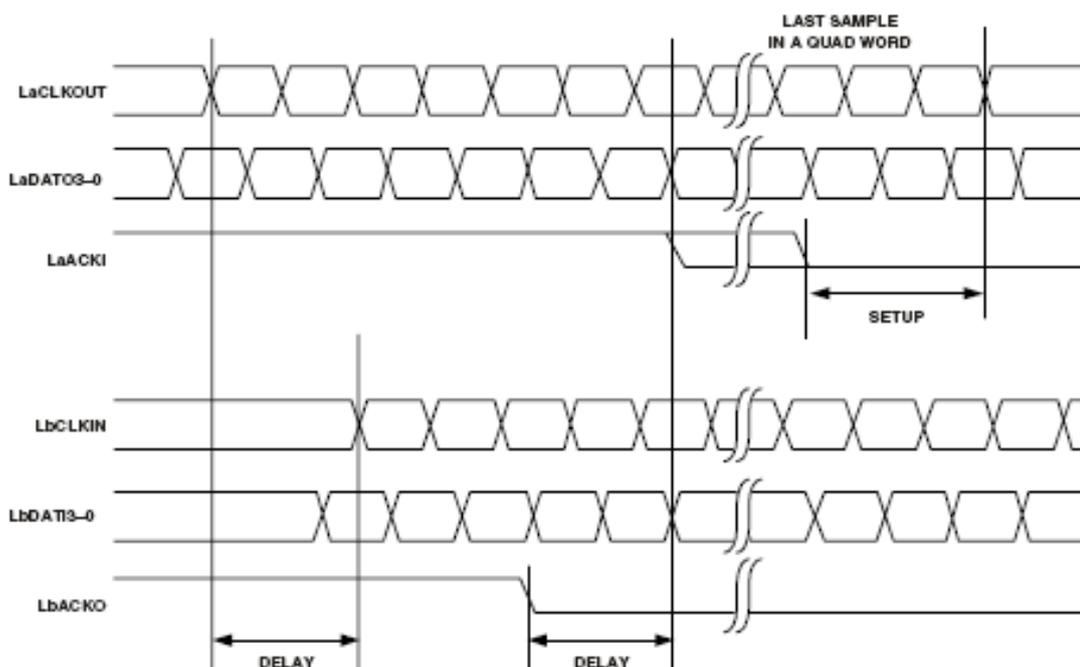


Рисунок 45 – Порт связи “а” передает порту связи “b” с задержкой

## 12.11 Механизмы определения ошибок порта связи

Порты связи поддерживают функции обнаружения ошибок в процессе работы. Когда порт связи обнаруживает состояние ошибки, он продолжает работу в соответствии со следующим алгоритмом:

- Если обнаружена ошибка истечения времени, устанавливаются соответственно разряды TTER (ошибка истечения времени передачи) в регистр LTSTAT или RTER (ошибка истечения времени приема) в регистр LRSTAT. Если ошибки истечения времени разрешены в регистрах LRCTL или LTCTL, активируется аппаратное прерывание.
- Если обнаружено переполнение данных в приемнике, разряд ROVER устанавливается в LRSTAT. Если разрешена ошибка переполнения данных приема в LRCTL, активируется аппаратное прерывание.
- Если верификация разрешена в LRCTL и обнаружена неверная контрольная сумма, разряд RCSER устанавливается соответственно в LRSTAT и активируется аппаратное прерывание.
- Если активное значение записывается в LRCTL или LTCTL и значение в регистре управления является активным, перезапись регистра предотвращается, разряд RWER устанавливается в LRSTAT или разряд TWER устанавливается в LTSTAT соответственно, активируется аппаратное прерывание.

Поля в LRSTAT и LTSTAT, указывающие на ошибку, могут быть очищены с помощью чтения из соответствующих очищенных регистров LRSTATC и LTSTATC.

### 12.11.1 Время ожидания передачи через порт связи

Если в передатчике LxASKI деактивирован в течение периода 2048 тактовых циклов и в передатчике есть данные, готовые к передаче, происходит ошибка времени ожидания (time-out) и разряд TTER устанавливается в LTSTAT. На линии LxASKI имеется слабый резистор, понижающий логический уровень. В случае если нет подключения приемника, вход будет в неактивном состоянии. Данная ошибка позволяет обнаружить отсутствие приемника вообще или сбой в его работе. Также эта ошибка позволяет предотвратить возможное зависание системы в случае ожидания готовности передачи.

### 12.11.2 Время ожидания приемника

Если принимающий буфер приемника полон и нет доступа для чтения в течение 2048 тактовых циклов, происходит ошибка окончания времени ожидания и разряд RTER устанавливается в LRSTAT. Данная ошибка информирует процессор о возможной некорректной работе программы обслуживания канала связи.

### 12.11.3 Ошибка верификации порта связи

Передатчик имеет возможность формировать байт контрольной суммы и отправлять его вместе с передаваемым счетверенным словом. Передача контрольного байта разрешается специальным битом регистра управления. Контрольный байт рассчитывается как сумма всех байтов данных, которые были переданы. 128-разрядное слово состоит из 16 байтов и контрольный байт отправляется в конце передачи каждые 16 байтов. Приемник принимает первые 16 байт данных, производит вычисление собственной контрольной суммы и сравнивает ее с принятым значением. Если два байта различаются, разряд RCSEr устанавливается в регистр LSTAT и формируется аппаратное прерывание.

Алгоритм контрольной суммы следующий:

–  $Checksum = \text{младший байт от суммы } (B_0 + B_1 + \dots + B_{15})$ .

### 12.11.4 Ошибка записи приема/передачи через порт связи

Запись активного значения в регистр управления приемника или передатчика является допустимой только при выключенном приемнике или передатчике. Запись активного значения в активный управляющий регистр вызывает ошибку записи. Таким образом для того, чтобы поменять некоторые параметры обмена необходимо предварительно выключить соответствующий приемник или передатчик и затем включить его снова с требуемым значением.

## 12.12 Регистр управления приемником порта связи (LRCTLx)

Регистры LRCTLx определяют параметры приема для порта связи. Подробное описание разрядов регистра приведено ниже (Таблица 78).

**Таблица 78 – Регистр LRCTL**

<b>Бит</b>	<b>Имя</b>	<b>Назначение</b>
0	REN	Бит включения приемника: 1 – включен 0 – выключен
1	RVERE	Разрешение контроля при приеме 1 – разрешено 0 – запрещено
2	RTOE	Разрешение прерывания в случае обнаружения ошибки time out
3	RBCMPE	Разрешение анализа входа LxBCMPI : 1 – разрешено 0 – запрещено
5:4	RDSIZE	Размер шины данных приема: 11 – 16 бит 10 – 8 бит 01 – 4 бита 00 – 1 бит
6	ROVRE	Разрешение прерывания при переполнении буфера приемника: 1 – разрешено 0 – запрещено
7	RINIF	Разрешение проведения процедуры инициализации: 1 – выполнить инициализацию с помощью RINIV значения 0 – не выполнять инициализацию
8	RINIV	Значение данных которые используются при процедуре инициализации
9		Разрешение генерации клона GPS
10		
11		Выбор приемника данных: 0 – буфер порта связи 1 – модуль UP-DOWN
13:12		Режим для ADC 8 бит шины данных
14		Режим представления данных
15		Режим ADC DDR
16		Источник клона: 0 – L0CLKI, 1 – L1CLKI
31:17	-	Зарезервировано

Регистр не может изменяться в процессе выполнения операции обмена. Запись активного значения в контрольный регистр допускается, только когда регистр имеет неактивное значение (REN сброшен). С целью изменения настроек во время работы порта связи, должно быть записано неактивное значение, чтобы прекратить работу порта, после чего следует записать новое активное значение. Игнорирование данного правила может вызвать аппаратное прерывание ошибки и новое значение не будет записано.

Выбор разрядности шины данных для обмена осуществляется в соответствии со следующим правилом:

- если бит 2 регистра CFG\_APB модуля CMU установлен в 1, разрядность шины данных приемников портов связи равна 4 бита;
- если бит 2 регистра CFG\_APB модуля CMU сброшен в 0, разрядность шины данных приемников портов связи определяется согласно значению бит RDSIZE.

## 12.13 Регистр управления передатчиком порта связи (LTCTLx)

Регистр устанавливает параметры передачи для порта связи. Значением сброса для регистров LTCTLx является 0. Подробное описание разрядов регистра приведено ниже (Таблица 79).

Таблица 79 – Регистр LTCTL

Бит	Имя	Назначение
0	TEN	Бит включения передатчика: 1 – включен 0 – выключен
1	TVERE	Разрешение формирования контрольной суммы при передаче: 1 – разрешено 0 – запрещено
2	TTOE	Разрешение прерывания в случае обнаружения ситуации time out
3	TBCMPE	Разрешение формирования выхода LxBCMPO: 1 – разрешено 0 – запрещено
5:4	TDSIZE	Размер шины передачи: 11 – 16 бит 10 – 8 бит 01 – 4 бита 00 – 1 бит
8:6	SPD	Частота передачи LxCLKOUT равна: 000 – CCLK 001 – CCLK деленный на 1.5 010 – CCLK деленный на 2 100 – CCLK деленный на 4 Другие значения зарезервированы
9		Источник клона передатчика
10		Инверсия внешнего клона
11		Выбор источника данных для передачи: 0 – передатчик порта связи 1 – модуль UP-DOWN
12		Режим представления данных
13		Режим внешнего DAC
31:14	-	Зарезервировано

Регистр не может изменяться в процессе выполнения операции обмена. Запись активного значения в контрольный регистр допускается, только когда регистр имеет неактивное значение (TEN сброшен). С целью изменения настроек во время работы порта связи, должно быть записано неактивное значение, чтобы прекратить работу порта, после чего следует записать новое активное значение. Игнорирование данного правила может вызвать аппаратное прерывание ошибки и новое значение не будет записано.

## 12.14 Регистр состояния приемника порта связи (LRSTATx)

Регистры LRSTATx указывают статус приемника порта связи. Значением сброса для регистров LRSTATx является 0x0000 0000. Подробное описание разрядов регистра приведено ниже (Таблица 80).

Таблица 80 – Регистр LRSTAT

Бит	Имя	Назначение
1:0	RSTAT	Состояние буфера приемника: 00 – пуст 01 – значение буфера достоверно 11 – буфер полон 10 – буфер не готов (принимает данные)
2	RTER	1 – приемник обнаружил ситуацию time out. 0 – нет ошибки
3	RWER	1 – приемник обнаружил ошибку записи 0 – нет ошибки
4	RCSER	1 – обнаружена ошибка контрольной суммы 0 – нет ошибки
5	ROVER	1 – ошибка переполнения приемника 0 – нет ошибки
6	RINIT	Состояние инициализации приемника: 1 – проинициализирован 0 – не инициализировался
31:7	-	Всегда ноль

## 12.15 Регистр состояния передатчика порта связи (LTSTATx)

Регистры LTSTATx указывают статус передатчика порта связи. Значением сброса регистров LTSTATx является = 0x0000 0002. Подробное описание разрядов регистра приведено ниже (Таблица 81).

Таблица 81 – Регистр LTSTAT

Бит	Имя	Назначение
0	TVACANT	1 – буфер может принять данные 0 – буфер не может принять данные
1	TEMPY	1 – передатчик пуст 0 – передатчик занят Бит может быть использован при выключении передатчика.
2	TTER	1 – ошибка передачи time out 0 – нет ошибки
3	TWER	1 – ошибка записи 0 – нет ошибки Активное значение было записано в регистр управления во время когда передатчик был включен.
31:4	-	Всегда ноль

## 13 Последовательный асинхронный интерфейс UART

Процессор имеет два интерфейса UART. Интерфейс имеет гибкую систему программирования, позволяющую задать различные скорости обмена, длину посылки, контроль четности, а также различные ситуации прерываний. Интерфейс имеет встроенные FIFO глубиной 8 байт для приема и передачи. Возможности интерфейса позволяют ему работать с контроллером DMA.

Интерфейсы имеют различные базовые адреса в карте памяти процессора: UART\_0 – **0x80000100**, UART\_1 – **0x80000120**. Каждый из интерфейсов имеет набор регистров, приведенный ниже (Таблица 82).

Таблица 82 – Регистры интерфейса

Номер регистра	Обозначение регистра	Описание	Состояние по сбросу
0	UARTDR	Регистр данных	-
1	RXSTAT	Регистр состояния приемника	0
2	UCR_load	Регистр управления загрузка	0
3	UCR_set	Регистр управления установка	0
4	UCR_clear	Регистр управления сброс	0
5	UBitRate	Регистр управления скоростью обмена	0
6	UFLAG	Регистр флагов	-
7	UINTM	Регистр запросов прерываний (маскированный)	0
8	UINT	Регистр запросов прерываний (немаскированный)	0

Условная структурная схема интерфейса приведена ниже (Рисунок 46). Регистры интерфейса позволяют пользователю задать требуемый режим работы: формат посылки, скорость обмена, источник прерываний и др. Рассмотрим более подробно назначение регистров интерфейса.

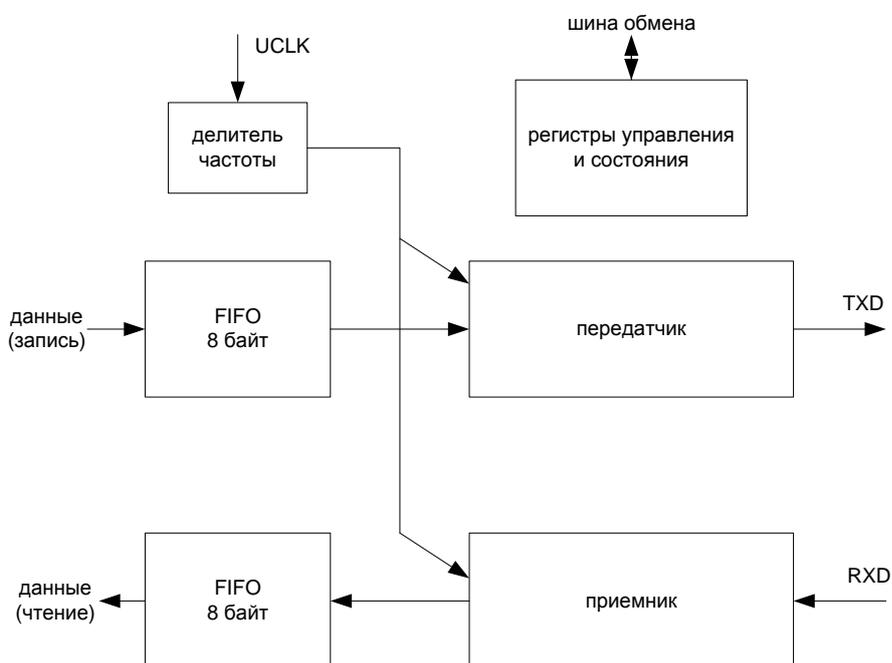


Рисунок 46 – Структурная схема UART

### 13.1 Регистр данных UARTDR

Регистр *UARTDR* представляет собой буфер данных передатчика при записи и буфер данных приемника при чтении. В качестве такого буфера используется FIFO емкостью 8 байт. Это позволяет выполнить одновременную загрузку 8 байт данных в буфер передатчика или обслуживать приемник только в случае заполнения буфера приемника не менее, чем наполовину. Это сокращает количество прерываний процессора на обслуживание интерфейса. Буфер приемника содержит не только байты данных, но и соответствующие им признаки ошибок приема-передачи. Во время чтения байта данных, происходит копирование соответствующих ему признаков в регистр состояния *RXSTAT*. Несмотря на то, что интерфейс подключен к 32-разрядной шине, при обмене данными используются только 8 младших бит шины.

### 13.2 Регистр состояния RXSTAT

**Таблица 83 – Регистр RXSTAT**

Бит	Обозначение	Тип	Описание
0	FRAME	R	Ошибка в формате посылки 0 – нет ошибки 1 – ошибка
1	PARITY	R	Ошибка четности 0 – нет ошибки 1 – ошибка
2	OVERRUN	R	Переполнение буфера приемника 0 – нет переполнения 1 – переполнение. Бит очищается после чтения регистра.
3	ERROR	R	Устанавливается в 1, если одна из выше описанных ошибок обнаруживается.
7:4		-	Не используется

Бит *FRAME* устанавливается в 1, если во время приема не был обнаружен Стоп-бит, т.е. посылка не завершена корректно. Бит *PARITY* устанавливается в 1, если в принятом байте обнаружена ошибка четности\нечетности. Бит *OVERRUN* устанавливается в 1, если после приема очередного байта обнаруживается, что буфер приемника уже полон и нет места для записи принятого байта. В этом случае происходит потеря принятого байта и установка признака переполнения. Если после чтения текущего байта данных и его регистра состояния обнаруживается, что бит переполнения, установлен, то еще, по крайней мере, 7 байт данных в буфере являются достоверными.

### 13.3 Регистры управления скоростью обмена UBitRate

Интерфейс содержит в себе программируемый делитель частоты обмена разрядностью 16 бит.

**Таблица 84 – Регистр UBitRate**

Бит	Обозначение	Тип	Описание
15:0	BR	R/W	Биты делителя

Входной частотой интерфейса является внешний вход синхронизации ХТІ. Внутри интерфейса она может дополнительно делиться на величину, заданную в регистре управления скоростью обмена, увеличенную на 1. Так, если значение равно 0x03, делитель будет делить входную частоту на 4. Эта частота будет использована приемником для сканирования принимающей линии. При этом предполагается, что длительность одного бита принимаемой посылки определяется битом UHBRE регистра UCR и может составлять 4 или 16 тактов базовой частоты. Таким образом, реальная скорость приема – передачи по интерфейсу равна  $(ХТІ / (U\mathit{BitRate} + 1)) / (16 \text{ или } 4)$ .

### 13.4 Регистр управления интерфейсом UCR

**Таблица 85 – Регистр UCR**

Бит	Обозначение	Тип	Описание
0	UARTEN	R/W	Разрешение работы UART интерфейса 0 – выключен 1 – включен
1	UTXDIS	R/W	Управление передатчиком 0 – включен 1 – выключен
2	URXDIS	R/W	Управление приемником 0 – включен 1 – выключен
3	DMAONERR	R/W	Управление запросом к контроллеру DMA во время обнаружения ошибки приема 0 – обнаружение ошибки не влияет формирование запроса к контроллеру DMA на обслуживание. 1 – формирование запроса к контроллеру DMA на обслуживание запрещено при обнаружении ошибки.
4	UTXFDIS	R/W	Управление буфером передатчика 0 – FIFO включено 1 – FIFO выключено
5	URXFDIS	R/W	Управление буфером приемника 0 – FIFO включено 1 – FIFO выключено
7:6	UHBRE	R/W	Управление длительностью одного бита посылки 00 - 1/16 clock 01 – 1/4 clock 1x – 1/4 clock, выключение внутреннего делителя
8	BREAK	R/W	Если равен 1, линия TXD передатчика устанавливается в высокий уровень
9	PRTEN	R/W	Управление проверкой и формированием бита четности 0 – запрещено 1 – разрешено
10	EVENPRT	R/W	Выбор проверки на четность\нечетность 0 – нечетность 1 – четность
11	XSTOP	R/W	Количество Стоп-битов (передача) 0 – один стоп-бит 1 – два стоп-бита приемник не проверяет количество принятых стоп-бит.
12	UFIFOEN	R/W	Управление буфером FIFO 0 – выключено (глубина буфера равна 1 слову)

			1 – включение FIFO приемника и передатчика
14:13	WRDLEN	R/W	Выбор длины передаваемых данных 00 – 8 bits 01 – 7 bits 10 – 6 bits 11 – 5 bits
15	INV	R/W	Инверсия TXD линии когда 1
16	TXINT	R/W	Управление разрешением прерывания от флага TXINT 0 – запрещено 1 – разрешено
17	RXINT	R/W	Управление разрешением прерывания от флага RXINT 0 – запрещено 1 – разрешено
18	RXERRINT	R/W	Управление разрешением прерывания от флага RXERRINT 0 – запрещено 1 – разрешено
19	MSINT	R/W	Управление разрешением прерывания от флага MSINT 0 – запрещено 1 – разрешено
20	UDINT	R/W	Управление разрешением прерывания от флага UDINT 0 – запрещено 1 – разрешено
21	UTXEINT	R/W	Управление разрешением прерывания от флага UTXEINT 0 – запрещено 1 – разрешено
22	URXTINT	R/W	Управление разрешением прерывания от флага URXTINT 0 – запрещено 1 – разрешено
30:23	RSV	-	Не используется
31	LBM	R/W	Тестовый режим работы интерфейса при котором выход передатчика коммутируется на вход приемника (когда ==1)

Бит UARTEN разрешает работу интерфейса. Он используется одновременно с битами UTXDIS и URXDIS, которые индивидуально управляют передатчиком и приемником соответственно.

Бит DMAONERR позволяет управлять интерфейсом во время его работы с контроллером DMA. В случае если при приеме данных возникнет ошибка, то при установленном бите DMAONERR произойдет блокировка запроса к контроллеру DMA, что исключит прием ошибочных данных.

Бит UTXFDIS используется только для управления выключением FIFO передатчика. При установленном бите FIFO выключен.

Бит URXFDIS используется только для управления выключением FIFO приемника. При установленном бите FIFO выключен.

Биты UHBRE определяют количество тактов частоты синхронизации (полученной после деления входной частоты блока UART на значение UBRCR) необходимых для приема (передачи) одного бита данных. Значение 00 задает режим, в котором длительность одного бита передаваемой (принимаемой) информации равна 16 периодам частоты внутреннего делителя. Этот режим

обеспечивает надежный прием в условиях помех. Значения 01, 10, 11 задают режим, в котором длительность одного бита передаваемой (принимаемой) информации равна 4 периодам частоты внутреннего делителя. В случае значений 10 или 11, значение коэффициента внутреннего делителя частоты равно 1 вне зависимости от значения в регистре **UBitRate**.

Таким образом, реальная скорость обмена интерфейса UART определяется как входная частота ХТІ, деленная на запрограммированный коэффициент в регистре **UBitRate**, затем деленная на 16 или 4 в передатчике (приемнике) интерфейса. Максимальная скорость обмена составляет ХТІ /4.

Бит BREAK. Установка данного бита приводит к установке высокого уровня на линии TXD передатчика.

Бит PRTEN разрешает (если равен 1) автоматическую генерацию дополнительного бита паритета при передаче байта данных. При приеме, установка этого бита разрешает проверку паритета принятого байта данных и генерацию флага об ошибке.

Бит EVENPRT позволяет задать тип проверки: четное или нечетное количество единичных бит в посылке. Если этот бит равен нулю, дополнительный бит четности дополняет количество единичных бит в посылке до четного количества. К примеру, если передается значение 0x15, бит четности равен 1. Если передается значение 0x55, то бит четности равен 0. Соответственно, если значение бита EVENPRT равно нулю, осуществляется дополнение до нечетного количества единичных бит.

Бит XSTOP позволяет задать количество Стоп-битов в одиночной посылке. Значение 0 соответствует одному стоп-биту, а значение 1 – двум стоп-битам. Этот бит используется только передатчиком. Приемник не проверяет количество принятых стоп-битов больше одного. Увеличение количества стоп-битов до двух иногда полезно в случаях, когда существует расхождение в скорости обмена передающей и принимающей сторон.

Бит UFIFOEN позволяет управлять буферами приемника и передатчика. Если этот бит, равен 0, флаги состояния буферов формируются так, как будто размер буфера равен одному байту. При установленном бите, размер буфера равен 8.

Биты WDRLEN позволяют запрограммировать количество реальных информационных бит в посылке. Таким образом, посылка может быть длиной от 7 бит (старт, 5 бит данных, стоп) до 12 бит (старт, 8 бит данных, бит четности и два стоповых).

### 13.5 Регистр состояния интерфейса UFLAG

**Таблица 86 – Регистр UFLAG**

Бит	Обозначение	Тип	Описание
0			
1			
2			
3	UBUSY	R	Передатчик занят Бит равен 1 во время передачи данных, включая стоп-биты
4	URXFE	R	Состояние буфера приемника. 0 – буфер приемника хранит данные 1 – буфер приемника пуст
5	UTXFF	R	Состояние буфера передатчика 0 – буфер приемника может принять новые данные 1 – буфер приемника заполнен полностью
7:6			

Бит UBUSY показывает состояние передатчика в текущий момент времени. Единичное значение бита указывает на процесс передачи информации. Флаг может быть использован для определения момента, когда передатчик полностью закончил работу.

Бит URXFE является признаком состояния FIFO приемника. Значение флага не зависит от того включено FIFO или нет. Единичное значение флага указывает на то, что буфер приемника пуст.

Бит UTXFF является признаком состояния FIFO передатчика. Значение флага зависит от того включено FIFO или нет. Если FIFO выключено, единичное значение флага указывает на то, что в буфере передатчика имеется байт данных для передачи. Если FIFO включено, единичное значение флага указывает на то, что буфер передатчика заполнен полностью.

### **13.6 Регистры управления прерываниями UINT, UINTM**

Интерфейс имеет регистр состояния запросов прерывания UINT. Каждый бит регистра состояния имеет соответствующий бит разрешения прерывания. Каждый из интерфейсов системы имеет только один вход запроса прерывания в контроллере прерываний. Поэтому на программиста возлагается задача определения, какая именно ситуация вызвала запрос на прерывание.

**Таблица 87 – Регистр UINT**

<b>Бит</b>	<b>Обозначение</b>	<b>Тип</b>	<b>Описание</b>
0	TXINT	R	Запрос прерывания от передатчика
1	RXINT	R	Запрос прерывания от приемника
2	RXERRINT	R	Ошибка приема во время работы с контроллером DMA
3	MSINT	R/W	Запрос прерывания от модема
4	UDINT	R	Запрос прерывания от приемника в состоянии «выключен»
5	UTXEINT	R/W	Запрос прерывания от буфера передатчика при отсутствии в нем данных
6	URXTINT	R/W	Запрос прерывания от приемника при обнаружении ситуации “time-out”
7	RSV	-	Не используется

Бит TXINT – запрос прерывания от передатчика. Он отражает состояние FIFO передатчика. Если FIFO выключено, запрос на прерывание активен (равен 1) если буфер пуст. Если же FIFO включено, запрос на прерывание активен, если буфер заполнен на половину или менее.

Бит RXINT – запрос прерывания от приемника. Запрос может быть активен в следующих случаях: FIFO приемника выключено и в нем имеются принятые данные, FIFO приемника включено и заполнено принятыми данными на половину или больше, FIFO приемника содержит данные и не поступило новых данных за время более чем период, равный передаче 32 бит данных.

Бит RXERRINT отражает информацию о возможных ошибках при приеме данных. Наиболее эффективно данный бит может быть использован при работе интерфейса с контроллером DMA. В этом случае, если контроллер DMA прочитает данные, при приеме которых были обнаружены ошибки, то интерфейс автоматически выработает прерывание, что позволит процессору правильно обработать возникшую ситуацию. Данный бит автоматически блокирует генерацию

запроса к контроллеру DMA на обслуживание (при разрешении в регистре управления).

Бит MSINT отражает запрос на прерывание от линий состояния модема. Изменение состояния любого из входов DCD, DSR, CTS вызывает установку бита MSINT.

Бит UDINT позволяет определить ситуацию, когда приемник интерфейса выключен, но на линии приемника RXD обнаружен старт бит. Это приводит к установке запроса на прерывание.

Бит UTXEINT активен в случае, когда интерфейс включен и буфер передатчика пуст.

Бит URXTINT отражает наличие ситуации в приемнике, когда нет приема за период равный передаче 32 бит данных. Этот флаг может быть использован для определения ситуации конца пакета данных.

Биты регистра состояния, помеченные как R/W, могут быть сброшены программно. Это осуществляется записью 1 в соответствующий бит регистра состояния. Запись нуля не изменяет значения бита.

Необходимо отметить следующую деталь – при чтении регистра UINTM происходит чтение значения регистра состояния, маскированное значением разрешения. Для чтения оригинального значения регистра состояния используется регистр UINT.

### **13.7 Программирование интерфейса UART**

Перед началом работы с интерфейсом UART необходимо разрешить альтернативную функцию для соответствующих битов портов общего назначения. Необходимо заранее установить требуемую скорость обмена и значения соответствующих коэффициентов деления в конкретном интерфейсе. После программирования скорости обмена необходимо задать другие параметры обмена (длину посылки, контроль ошибок, ситуации прерываний) и включить интерфейс. Интерфейсы UART\_0, UART\_1 могут работать с контроллером DMA.

## 14 Последовательный синхронный интерфейс SPI

Интерфейс включает сигнал синхронизации, разрешения обмена, вход и выход данных. Он имеет гибкую систему программирования, позволяющую задать различные скорости обмена, длину посылки, а также различные ситуации прерываний. Интерфейс имеет встроенные FIFO глубиной 8 байт для приема и передачи. Возможности интерфейса позволяют ему работать с контроллером DMA.

Для обмена посредством интерфейса SPI используются четыре линии: синхронизация, вход данных, выход данных, разрешение обмена. В ряде случаев используется еще дополнительная линия выбора мастера-подчиненного, однако, в данной реализации такой выбор осуществляется посредством программирования регистров управления. Направление линий интерфейса, в зависимости от выбранного режима работы, приведено ниже (Таблица 88):

**Таблица 88 – Режимы работы интерфейса**

	SPI (Ведущий)	SPI (Ведомый)
SPI_CLK	Выход	Вход
SPI_DO	Выход	Вход
SPI_DI	Вход	Выход
SPI_EN	Выход	Вход

Интерфейс доступен для программирования как набор регистров, позволяющих задать нужную конфигурацию и осуществить прием-передачу данных. Регистры интерфейса подключены к шине обмена периферийных устройств и имеют базовый адрес **0x80000140**. Краткая информация о регистрах приведена ниже (Таблица 89).

**Таблица 89 – Регистры интерфейса**

номер	Имя	Назначение	По сбросу
0	SPCR0	Регистр управления 0	0
1	SPCR1	Регистр управления 1	0
2	SPDR	Регистр данных	-
3	SPSR	Регистр состояния	0
4	RX_CNT	Счетчик приемника	0

Алгоритм работы интерфейса предполагает одновременный прием и передачу данных. Интерфейс имеет встроенные буфера (FIFO) для приема и передачи данных. Размер каждого из буферов составляет восемь 16-разрядных слов. Для того чтобы осуществить обмен посредством интерфейса, необходимо:

- запрограммировать параметры обмена в регистрах управления;
- записать данные для передачи в буфер передатчика;
- прочитать принятые данные из буфера приемника.

Если необходимо только передать данные в какое-то устройство, интерфейс все равно будет принимать данные с входной линии данных. Поэтому после окончания передачи необходимо сбросить флаги состояния приемника.

Если необходимо только принять информацию, в этом случае нужно записать в буфер передатчика фиктивные данные для передачи. В этом случае количество принятых данных будет соответствовать количеству переданных.

Интерфейс имеет возможность подключать до 6-ти внешних устройств. Каждое из устройств имеет общие линии синхронизации и данных, но индивидуальную линию выбора устройства. Номер устройства задается в регистре управления и должен быть неизменным на весь период обмена. Имеется возможность индивидуального задания неактивного уровня для каждого устройства.

Если возникает необходимость работы интерфейса в режиме «подчиненный», ведущее устройство должно осуществлять выборку интерфейса посредством входа CS1. Также если выбран вариант начальной загрузки с использованием SPI-флэш памяти, данная память должна быть подключена к CS0.

Практически количество подключаемых внешних SPI-устройств может быть и больше чем 6. В этом случае в качестве сигнала выбора устройства можно использовать любой из свободных выходов общего назначения, а в интерфейсе при осуществлении обмена задавать номер устройства 7.

Интерфейс имеет два 16-разрядные регистра управления, позволяющие задать параметры работы интерфейса. Ниже, в таблицах, приведено краткое описание разрядов регистров управления.

## 14.1 Регистр SPCR0

**Таблица 90 – Регистр управления SPCR0**

Бит	Имя	Тип	Назначение
3:0	DSS	R/W	Выбор размера данных 0000 - Не используется 0001 - Не используется 0010 - Не используется 0011 - 4-бит данные 0100 - 5- бит данные 0101 - 6- бит данные 0110 - 7- бит данные 0111 - 8- бит данные 1000 - 9- бит данные 1001 - 10- бит данные 1010 - 11- бит данные 1011 - 12- бит данные 1100 - 13- бит данные 1101 - 14- бит данные 1110 - 15- бит данные 1111 - 16- бит данные
4	-		
5	-		
6	SPO	R/W	Полярность SPI_CLK 0 – SPI_CLK высокий импульс 1 – SPI_CLK низкий импульс
7	SPH	R/W	Фаза SPI_CLK 0 – SPI_CLK активен на заднем фронте 1 – SPI_CLK активен на переднем фронте
8	TWI_on	R/W	0 – стандартный SPI интерфейс 1 – 3-х проводной интерфейс с двунаправленной линией

9	TWI_rw	R/W	Чтение(0) или запись(1) в режиме TWI
10	LMSF	R/W	Выбор 0 – старший бит первый 1 – младший бит первый
11	-		
14:12	CSN		Выбор номера SPI-устройства
15	-		
27:16	SCR	R/W	Коэффициент деления частоты Скорость обмена = SOC_CLK/ (SCR + 1)
31:27	-		

Биты DSS задают количество передаваемых бит в одной посылке. Максимальное количество бит в одной посылке равно 16. Если для работы интерфейса выбрана меньшая разрядность передаваемых данных (от 4 до 15), для передачи используются только младшие значащие биты. Так, если размер одиночной посылки равен 8 бит, старшие биты (с 8 по 15) слова не будут передаваться.

Биты SPO и SPH управляют полярностью и фазой синхросигнала SPI\_CLK. Ниже показаны возможные варианты обмена при передаче 4-х бит данных (Рисунок 47).

Диаграмма обмена SPI интерфейса при DSS = 3

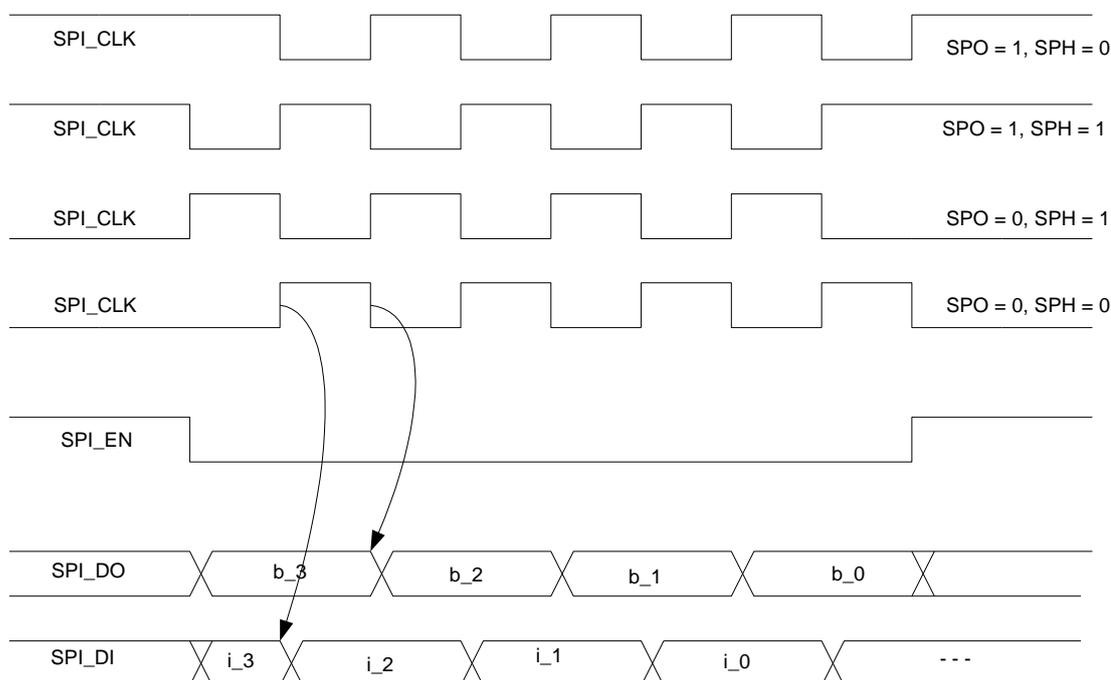


Рисунок 47 – Диаграмма обмена в режиме master

Биты SCR задают коэффициент деления для внутреннего делителя частоты. Входная частота синхронизации модуля SPI поступает с периферийной шины, затем она может дополнительно делиться на запрограммированный коэффициент деления согласно формуле:

$$SPI\_CLK = SOC\_CLK \div (SCR + 1).$$

## 14.2 Регистр SPCR1

Таблица 91 – Регистр управления SPCR1

Бит	Имя	Тип	Назначение
0	RIM	R/W	Управление прерыванием от приемника интерфейса 0 – запрещено 1 – разрешено
1	TIM	R/W	Управление прерыванием от передатчика интерфейс 0 – запрещено 1 – разрешено
2	LBM	R/W	Разрешение тестового режима работы 0 – обычный режим работы 1 – тестовый режим работы
3	SPE	R/W	Разрешение работы интерфейса 0 – выключен 1 – включен
4	MS	R/W	Режим работы 0 – мастер 1 – подчиненный
5	TUM	R/W	Разрешение прерывания по исчерпанию буфера передатчика при работе в режиме подчиненного 0 – запрещено 1 – разрешено
6	ROM	R/W	Разрешение прерывания при переполнении буфера приемника 0 – запрещено 1 – разрешено
7	-	R/W	
8	R_RQM	R/W	Запрос на обслуживание со стороны FIFO приемника: 0 – FIFO загружено на половину или более 1 – в FIFO имеются данные
9	T_RQM	R/W	Запрос на обслуживание со стороны FIFO передатчика: 0 – FIFO загружено на половину или менее 1 – FIFO не заполнено полностью
10	TXO	R/W	Только передача 0 – одновременный прием-передача 1 – только передача
11	R XO	R/W	Только прием 0 – одновременный прием-передача 1 – только прием
12	CIM	R/W	Разрешение прерывания от счетчика приемника 0 – запрещено 1 – разрешено
13			
14	HOLDCS	R/W	Удержание линии CS активной после завершения обмена 0 – после обмена линия становится неактивной. 1 – после очередного обмена линия остается активной.
15	ROTL		Выходной уровень на выходе данных в случае работы в режиме «только прием»
21:16	CSAL	R/W	Выбор активного уровня на выходе CSi 0 – активный низкий 1 – активный высокий

Бит SPE разрешает работу интерфейса. Если его значение равно нулю – интерфейс выключен. Для включения интерфейса необходимо записать в данные бит значение 1.

Бит MS при включенном интерфейсе определяет режим работы интерфейса – мастер или подчиненный. В случае работы в режиме мастер (MS = 0) интерфейс самостоятельно вырабатывает сигнал синхронизации и сигнал разрешения обмена. В случае работы в режиме «подчиненный» (MS = 1), интерфейс работает под управлением внешнего синхросигнала и от внешнего сигнала разрешения работы. При работе в режиме подчиненный имеются некоторые ограничения на максимальную частоту обмена. Она должна быть как минимум в три раза ниже частоты обмена по внутренней шине периферийных устройств.

Бит RIM управляет разрешением прерывания от приемника интерфейса. Если бит RFS регистра состояния установлен и бит RIM равен единице, это вызовет запрос прерывания.

Бит TIM управляет разрешением прерывания от передатчика интерфейса. Если бит TFS регистра состояния установлен и бит TIM равен единице, это вызовет запрос прерывания.

Бит LBM позволяет перевести интерфейс в тестовый режим работы (LBM = 1), в котором выход данных передатчика подается на вход данных приемника. Это позволяет произвести проверку оборудования без подключения внешнего устройства. Для нормального режима работы необходимо устанавливать LBM = 0.

Бит TXO позволяет, при необходимости, упростить передачу информации. Как уже отмечалось выше, при обычной работе интерфейс при передаче данных осуществляет одновременный прием информации с линии входа данных. Порой эти данные не нужны т.к. требуется выполнить только передачу. Но буфер приемника все равно заполняется. Чтобы исключить это неудобство был добавлен бит TXO, который при значении 1 выполняет блокировку приема данных и не изменяет состояние приемника.

Бит RXO позволяет, при необходимости, упростить прием информации. Как уже отмечалось выше, при обычной работе интерфейс для приема данных должен осуществить фиктивную передачу информации. Чтобы исключить это неудобство был добавлен бит RXO, который при значении 1 не требует наличия фиктивных данных в буфере передатчика при приеме данных. Для обеспечения функции «только прием» был добавлен дополнительный регистр RX\_CNT.

Бит CIM разрешает генерацию запроса прерывания при значении счетчика RX\_CNT равным нулю.

Бит CSAL позволяет задать активный уровень на выходе CS. Одни устройства могут активизироваться высоким уровнем, а другие – низким. Данный бит дает возможность задать активный уровень.

Бит HOLDCS позволяет осуществить удержание вывода CS в активном состоянии после завершения обмена. Бит должен быть установлен до начала обмена. После такого, как с началом обмена линия CS активизируется, она будет оставаться активной до тех пор, пока бит HOLDCS не будет сброшен. Сбросить бит можно во время осуществления последнего обмена или после его завершения. Использование данной функции позволяет осуществить передачи или прием посылки произвольной длительности.

### **14.3 Регистр счета принимаемых данных RX\_CNT**

Регистр используется только при установленным бите RXO==1, что соответствует режиму «только прием». В этом случае в регистр RX\_CNT необходимо записать количество принимаемых слов данных (не более 4095). Если

значение регистра не равно нулю, это автоматически формируется запрос на прием данных. После приема очередного слова осуществляется вычитание 1 из счетчика. Возможна генерация прерывания от счетчика в случае достижения им нулевого значения. Это происходит при установленном бите SIM.

#### **14.4 Регистр данных SPDR**

Регистр данных имеет разрядность 16 бит. При записи, данные попадают в верхний свободный регистр FIFO передатчика, а при чтении, считываются из нижнего регистра FIFO приемника. Размер FIFO приемника и передатчика составляет по восемь 16-разрядных слов каждый. Каждый из FIFO имеет указатели считывания и записи. Интерфейс имеет несколько флагов, которые отражают состояние FIFO передатчика и приемника. Интерфейс подключен к 32-разрядной шине периферийных устройств, однако при работе с регистром данных только младшие 16 бит имеют значение. Старшие 16 разрядов при записи данных игнорируются, а при чтении равны нулю.

#### **14.5 Регистр состояния SPSR**

**Таблица 92 – Регистр состояния SPSR**

<b>Бит</b>	<b>Имя</b>	<b>Тип</b>	<b>Назначение</b>
0	TFE	R	Наличие данных в буфере передатчика 0 – имеются 1 – отсутствуют
1	TNF	R	FIFO передатчика не заполнено 0 – FIFO передатчика заполнено полностью 1 – FIFO передатчика не заполнено
2	RNE	R	FIFO приемника имеет данные 0 – нет данных в FIFO приемника 1 – в FIFO приемника имеются данные
3	BSY	R	Занятость SPI интерфейса 0 – простаивает или выключен 1 – занят передачей-приемом
4	TFS	R	Запрос FIFO передатчика на обслуживание 0 – буфер передатчика более чем на половину заполнен (5 или более слов записано) 1 – буфер передатчика заполнен на половину или менее (4 или меньше слов записано)
5	RFS	R	Запрос FIFO приемника на обслуживание 0 – буфер приемника заполнен меньше чем на половину (3 или меньше слов записано) 1 – буфер приемника заполнен на половину или более (4 или больше слов записано)
6	ROR	R/W	переполнение FIFO приемника 0 – нет ошибки 1 – во время приема произошло переполнение буфера.
7	TUR	R/W	FIFO передатчика пусто (только в режиме «подчиненный») 0 – нет ошибки 1 – передатчик пытался передать данные из пустого буфера
8	RFF	R	FIFO приемника заполнено 0 – буфер приемника заполнен не полностью 1 – буфер приемника заполнен полностью
31:9	RSV	-	Не используются

Биты регистра SPSR отражают состояние интерфейса во время приема-передачи данных.

Бит TFE отражает наличие данных в буфере передатчика. Значение бита 1 соответствует ситуации, когда буфер пуст. Бит может быть использован для отслеживания ситуации, когда интерфейс закончил передачу.

Бит TNF может быть использован при заполнении буфера передатчика данными. Значение бита TNF=1 означает, что буфер может принять новые данные. Значение бита равно нулю означает что, буфер заполнен полностью.

Бит RNE отражает состояние буфера приемника. Значение бита RNE=1, означает наличие данных в буфере приемника, которые могут быть прочитаны.

Бит BSY отражает занятость интерфейса приемом\передачей данных. Бит может быть использован для определения момента полного завершения работы интерфейса. Так отсутствие данных в FIFO передатчика и равенство бита BSY нулю, означает завершение работы интерфейса.

Бит TFS используется как запрос передатчика на прерывание процессора или на обслуживание со стороны контроллера DMA. Равенство бита единице означает, что FIFO передатчика заполнено на половину или менее (т.е. в FIFO записано от 1 до 4-х слов).

Бит RFS используется как запрос приемника на прерывание процессора или на обслуживание со стороны контроллера DMA. Равенство бита единице означает, что FIFO приемника заполнено на половину или более (т.е. в FIFO записано от 4 до 8 слов).

Бит ROR отслеживает ситуацию переполнения FIFO приемника, т.е. случая, когда в буфер приемника была произведена попытка записи новых данных в то время, когда он был заполнен. Бит может вызвать запрос прерывания процессору, если соответствующий ему бит ROM регистра управления установлен.

Бит TUR имеет смысл при работе интерфейса в режиме подчиненного и отслеживает ситуацию, когда при запросе обмена со стороны мастера, буфер передатчика пуст. При работе в режиме подчиненного обмен инициирует внешний мастер, а данные должны быть загружены в буфер предварительно. Бит может вызвать запрос прерывания процессору, если соответствующий ему бит TUM регистра управления установлен.

Биты ROR и TUR устанавливаются аппаратно, но могут быть сброшены программно. Для очистки бита необходимо произвести запись в соответствующий бит значения 1.

Бит RFF отслеживает ситуацию, когда буфер приемника заполнен полностью.

## **14.6 Особенности работы в режиме “slave”**

Интерфейс имеет возможность работы в режиме подчиненного (“slave”) устройства. Данный режим задается при помощи установки в 1 бита MS регистра SPCR1. В данном режиме интерфейс использует вход SPI\_CLK (PA[4]) как вход синхронизации, вход SPI\_CS[1] (PA[8]) как вход выборки интерфейса, вход SPI\_DO (PA[5]) как вход данных для приема информации и выход SPI\_DI (PA[6]) для передачи данных ведущему устройству. В отличие от режима мастера, все внешние контакты интерфейса меняют направление на противоположное. Работа в режиме подчиненного устройства имеет ряд ограничений. Вход синхронизации SPI\_CLK используется, так как он есть на внешнем выводе, т.е. нет никаких преобразований полярности и фазы сигнала синхронизации. Выдача данных на SPI\_DI осуществляется только по отрицательному (из 1 в 0) фронту SPI\_CLK. Прием

данных с SPI\_DO выполняется только по положительному (из 0 в 1) фронту SPI\_CLK. Пример временной диаграммы представлен ниже (Рисунок 48).

Диаграмма обмена в режиме "slave" при DSS = 3

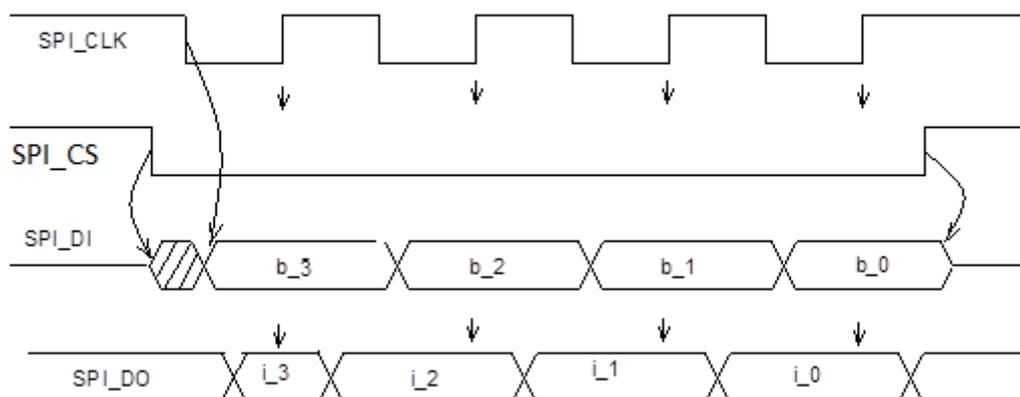


Рисунок 48 – Диаграмма обмена в режиме slave

Как следует из рисунка, выход SPI\_DI становится активным, как только на входе SPI\_CS появляется активный уровень. Активный уровень на входе SPI\_CS может программироваться. Если бит CSAL[1] равен нулю, активным уровнем на входе SPI\_CS будет низкий уровень. Если CSAL[1] равен единице – активный уровень высокий. Однако установка активного уровня на SPI\_CS не означает, что на выходе SPI\_DI сразу же появляются истинные данные. Только после первого отрицательного фронта на SPI\_CLK на выходе SPI\_DI появится достоверная информация (бит b\_3). Ведущее устройство должно принять данные с линии SPI\_DI по положительному фронту SPI\_CLK. В этот же момент и подчиненное устройство примет данные с линии SPI\_DO.

В момент приема бита b\_0 подчиненное устройство зафиксирует окончание приема посылки и передаст принятые данные в FIFO приемника. При передаче данных в FIFO будет выполнена передача из домена синхронизации SPI\_CLK в домен синхронизации SOC\_CLK. Для корректной работы интерфейса необходимо чтобы частота СОК-шины была как минимум в 3 раза выше частоты SPI\_CLK.

Поскольку момент начала передачи по интерфейсу в режиме подчиненного является случайным, данные для передачи должны быть записаны до момента начала передачи. В противном случае будет сформирован флаг ошибки. Моментом захвата данных из FIFO передатчика является первый положительный фронт сигнала SPI\_CLK. К этому моменту данные для передачи должны быть записаны в FIFO. Сброс указателя на взятые из FIFO данные выполняется немного позже. Возможна ситуация когда данные захвачены некорректно, но флаг ошибки не установится. Это может случиться, если одновременно с записью данных в FIFO приходит первый положительный фронт SPI\_CLK.

Прием-передача данных в подчиненном устройстве выполняется в сдвиговом регистре тактируемым SPI\_CLK. Таким образом, для обеспечения высокой скорости обмена желательно использовать большую длину посылки. В этом случае у процессора будет больше времени для обслуживания одной порции принятых данных.

## 15 Контроллер видеокамеры

Данный контроллер позволяет организовать прием информации от внешней видеокамеры или иного устройства поддерживающего заданный интерфейс.

Интерфейс видеокамеры представляет собой контроллер, подключенный к шине периферийных устройств и способный принимать данные от внешней цифровой видеокамеры (или другого устройства) согласно заданному протоколу обмена. Принятые данные помещаются в буфер и могут быть переданы в запоминающие устройства процессора с помощью контроллера DMA или с помощью процессора.

### 15.1 Регистры интерфейса

Регистры интерфейса подключены к шине обмена периферийных устройств и имеют базовый адрес **0x800001A0**. Краткое описание регистров приведено ниже (Таблица 93).

Таблица 93 – Регистры интерфейса

Номер	Название	Назначение
0	VC_DR	128-разрядный регистр данных. Доступен только по чтению и является выходом FIFO
4	VC_SR	Регистр состояния
5	VC_CR	Регистр управления

#### 15.1.1 Регистр управления CR

Таблица 94 – Регистр управления

Бит	Имя	Тип	Назначение
0	VCON	R/W	Включение интерфейса. При записи 1 интерфейс начинает прием данных во внутреннее FIFO.
1	VCIE	R/W	Разрешение прерывания. При записи 1 интерфейс вырабатывает прерывание если во внутреннем буфере имеются данные
2	SMODE	R/W	Режим приема информации: 0 – видеокамера 1 – ведущее устройство

#### 15.1.2 Регистр состояния SR

Таблица 95 – Регистр состояния

Биты	имя	Описание
0	EMPTY	1 – нет данных в буфере 0 – имеются данные
1	FULL	0 – буфер заполнен не полностью 1 – буфер заполнен полностью
2	OVERF	1 – было переполнение буфера во время приема данных 0 – не было переполнения
3	UNDERF	1 – при чтении данных произошло чтение пустого буфера

### 15.1.3 Регистр данных DR

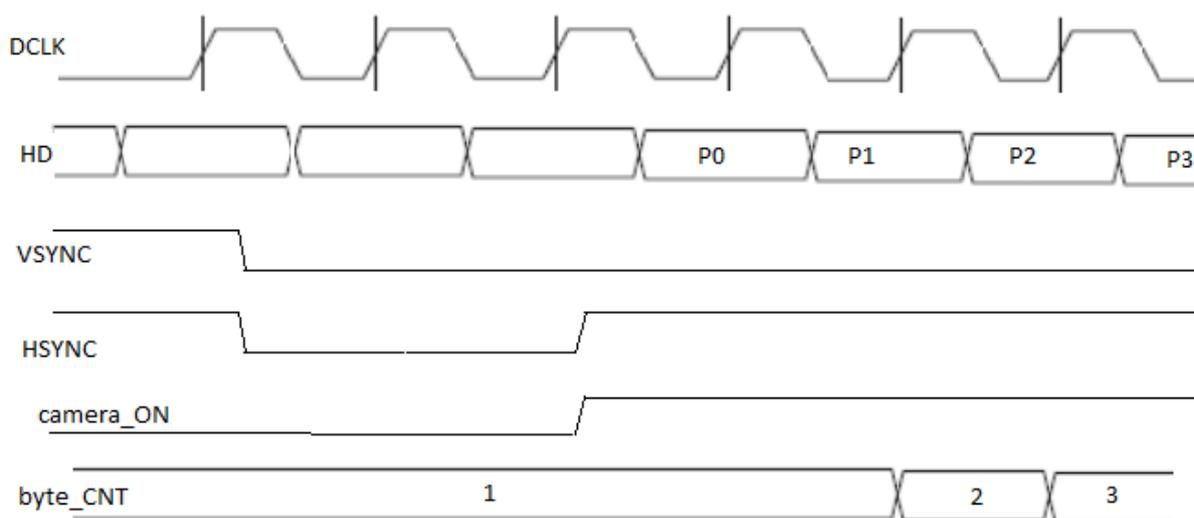
Посредством этого регистра происходит чтение данных после приема. Регистр имеет разрядность 128 бит и поэтому обращение к нему должно выполняться как к квадрослову. Размер буфера приема данных равен четырем 128-разрядным словам.

Интерфейс имеет внешние входы:

- DCLK – вход синхронизации;
- VSYNC – строб начала картины;
- HSYNC – строб начала линии;
- HD[7:0] – шина данных.

### 15.2 Режим приема «видеокамера»

Интерфейс является синхронным и осуществляет прием данных по положительному фронту сигнала синхронизации DCLK. Для начала приема информации, контроллер должен быть включен путем установки бита VCON регистра управления. После включения интерфейс начинает отслеживать стартовую ситуацию для начала приема данных. В случае режима видеокамеры для начала приема данных необходимо одновременное наличие на входах VSYNC и HSYNC высокого уровня. Этот факт запоминается во внутреннем флаге. В момент, когда оба входа VSYNC и HSYNC станут равным нулю (не обязательно одновременно как на рисунке), будет установлен внутренний флаг camera\_ON, означающий возможность приема данных. Временная диаграмма приведена ниже (Рисунок 49).



**Рисунок 49 – Начало приема данных в режиме «видеокамера»**

Прием данных начнется в момент, когда внутренний флаг camera\_ON будет установлен и на входе HSYNC будет высокий уровень, что означает наличие данных горизонтальной строки на входе контроллера. На временной диаграмме видно расположение позиции первого принимаемого байта (P0). Каждый раз, когда HSYNC равен 1 и включен прием, происходит наращивание счетчика байт byte\_CNT. Блок данных из 16 байт формирует квадрослово данных. Когда счетчик данных достигает значения 16 – квадрослово данных принято, формируется флаг готовности и принятая информация записывается в выходное FIFO. Счетчик опять начинает счет

сначала. Размер выходного буфера равен 4 квадрословам. При перезаписи данных в выходное FIFO выполняется передача информации из домена синхронизации DCLK в домен синхронизации SOC-шины. Как только в выходном FIFO появляются данные, формируется запрос к контроллеру DMA на обслуживание. К DMA может прочитать данные из буфера интерфейса и переслать во внутреннюю или внешнюю память (в соответствии с настройками канала DMA). В канале контроллера DMA происходит отслеживание объема переданной информации и при необходимости может быть сформирован запрос на прерывание для анализа принятой информации. Если контроллер DMA не успевает принимать поток данных от интерфейса, возможно переполнение буфера. В этом случае будет установлен соответствующий флаг в регистре состояния.

### 15.3 Режим приема «ведущее устройство»

К внешним контактам интерфейса видеокамеры может быть подключено некоторое ведущее устройство, имеющее стандартный интерфейс микропроцессорной шины (адрес, данные, сигналы выборки и записи), с целью передачи информации по определенному протоколу. Интерфейс видеокамеры поддерживает простой вариант обмена, удобный для такого подключения. Данный вариант включается при установке бита SMOD в 1. В этом случае возможно следующее подключение сигналов:

- DCLK как сигнал выборки интерфейса nVCS;
- VSYNC как сигнал адресной шины A[1];
- HSYNC как сигнал адресной шины A[0];
- HD[7:0] как младший байт шины данных D[7:0].

Сигнал выборки nVCS должен быть сформирован как логическое «или» сигналов выборки и записи, т.е.  $nVCS = nCS \text{ or } nWE$  (активный уровень низкий). Дополнительно к сигналу выборки может подключаться дешифратор адреса для дополнительного выбора устройства. При таком подключении обычный цикл записи по указанному адресу вызывает загрузку байта данных в интерфейс контроллера. Пример временной диаграммы приведен ниже (Рисунок 50).

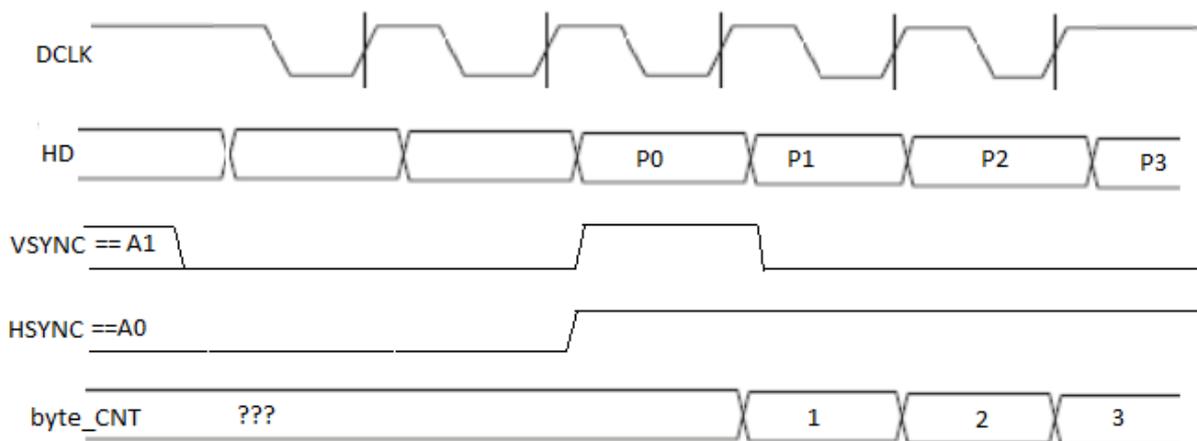


Рисунок 50 – Начало приема данных в режиме «ведущее устройство»

Для ведущего устройства передача данных в интерфейс выглядит как обычный цикл записи по адресам 0, 1, 2, 3. При этом:

- в интерфейс передаются только 8 бит данных;
- запись по адресу 0 выглядит как подача одного такта синхросигнала DCLK без выполнения каких-либо действий внутри интерфейса;
- запись по адресу 3 вызывает прием байта данных и установку счетчика принятых байт в 1. Это позволяет ведущему устройству определить начальное значение счетчика;
- запись по адресу 1 приводит к записи следующего байта и автоматическому увеличению счетчика принятых байт на 1.

После приема 16 байт выполняется передача квадрослова данных в выходное FIFO, и формируется запрос в контроллеру DMA или прерывание к процессору. Если передаваемый блок данных не кратен 16 байт, его необходимо дополнить недостающими байтами. При формировании сигнала записи (DCLK) необходимо обеспечить требуемое время предустановки входных сигналов и время удержания относительно положительного фронта stroba записи.

Таким образом, в режиме «ведущее устройство», интерфейс видеокамеры может рассматриваться как ведомый порт для приема информации от ведущего устройства. В зависимости от скорости передачи информации для обслуживания интерфейса может использоваться контроллер DMA или процессор.

## 16 Контроллер внешней NAND флэш-памяти

К процессору может быть подключена внешняя NAND флэш-память. Специальный контроллер позволяет осуществить преобразование запроса процессора на запись или чтение данных в последовательность команд обращения к флэш-памяти. Имеется поддержка коррекции ошибок.

При использовании данного контроллера разрядность шины данных контроллера внешней памяти уменьшается до 16 бит. Процессор не имеет прямого доступа во флэш-память, и все операции осуществляются посредством программирования операций контроллера.

Возможно подключение до трех внешних банков памяти. Выбор конкретного банка определяется разрядами 31:30 регистра адреса AR:

- 00 – NF\_CS 0;
- 01 – NF\_CS 1;
- 10 – NF\_CS 2.

Биты AR[29:0] позволяют прямо адресовать 4 Гбайт каждого банка памяти. Если подключаемая память имеет больший объем, можно использовать биты расширения ADDRH[2:0] регистра NAND\_CFG. Это позволяет адресовать до 32 Гбайт каждого банка флэш-памяти. Чтение и запись в NAND флэш-память имеет свои особенности. При описании протокола обмена с флэш-памятью, будет подразумеваться работа только(!) с флэш-памятью фирмы Samsung, а также памятью с аналогичным интерфейсом от других компаний.

Контроллер флэш-памяти имеет набор регистров, позволяющих задать параметры внешней подключенной памяти и режимы работы. Регистры контроллера доступны со стороны шины периферийных устройств и имеют базовый адрес 0x80000240.

С помощью регистров конфигурации пользователь может задать направление обмена (чтение или запись), начальный адрес обмена, количество передаваемых слов, а также размер слова (int, short, char). После задания необходимых параметров контроллер самостоятельно читает данные из внешней памяти во внутренний буфер (в случае чтения) или записывает данные из внутреннего буфера во внешнюю память (в случае записи). Для передачи данных между внутренним буфером контроллера и памятью системы, может использоваться процессор или контроллер прямого доступа. Ниже показано условно-графическое обозначение модуля контроллера флэш-памяти (Рисунок 51).

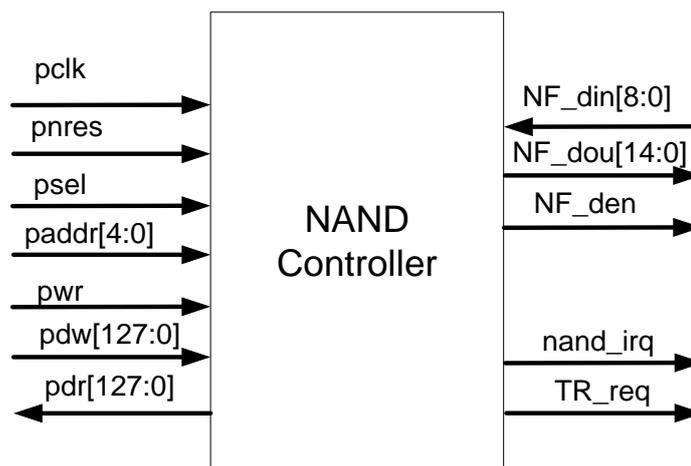


Рисунок 51 – Условно-графическое обозначение контроллера

На рисунке показаны интерфейс подключения контроллера к шине периферийных устройств, посредством которого пользователь может осуществлять программирование операций обмена, а также интерфейс для взаимодействия с внешней флэш-памятью. Подробнее описание сигналов модуля приведено ниже (Таблица 96).

**Таблица 96 – Сигналы контроллера флэш-памяти**

Сигнал	Направление	Активный уровень	Описание
<b>APB интерфейс</b>			
pclk	Вход	-	Тактовый сигнал
pnres	Вход	Низкий	Сигнал сброса
psel	Вход	Высокий	Сигнал выбора
paddr[4:0]	Вход	-	Адрес
pwr	Вход	-	Сигнал чтения или записи (низкий – чтение, высокий – запись)
pdw[127:0]	Вход	-	Данные на запись
pdr[127:0]	Выход	-	Прочитанные данные
<b>NAND интерфейс</b>			
NF_din[7:0]	Вход	-	Шина входных данных
NF_din[8]	Вход	-	Соответствует выходу Rb_n микросхемы – индикация занятости микросхемы (0 – занято)
NF_dou[7:0]	Выход	-	Шина выходных данных (команды, адреса)
NF_dou[8]	Выход	Высокий	CLE – признак команды
NF_dou[9]	Выход	Высокий	ALE – признак адреса
NF_dou[10]	Выход	Низкий	We_n – строб записи
NF_dou[11]	Выход	Низкий	Re_n – строб чтения
NF_dou[12]	Выход	Низкий	Ce_n – выборка микросхемы 0
NF_dou[13]	Выход	Низкий	Ce_n – выборка микросхемы 1
NF_dou[14]	Выход	Низкий	Ce_n – выборка микросхемы 2
NF_den	Выход	-	Сигнал, определяющий направление для двунаправленного порта DIO микросхемы флэш-памяти (1 – данные поступают во флэш, 0 – данные поступают из флэш). Пример: assign IO_bus = NF_den ? NF_dou[7:0] : 8'bz; assign NF_di[7:0] = IO_bus;
<b>Сигналы запросов</b>			
nand_irq	Выход	Высокий	Запрос на прерывание
TR_req	Выход	Высокий	Запроса на обмен с системным DMA

Контроллер может использоваться для совместной работы с контроллером DMA. Для этих целей предусмотрен выход TR\_req. Изменение выхода TR\_req из состояния 0 в состояние 1 сигнализирует контроллеру DMA о необходимости провести обмен данными.

Соответствие сигналов управления внешней флэш-памятью контактными площадкам микросхемы приведено ниже (Таблица 97).

**Таблица 97 – Внешние контакты контроллера**

Имя	Порт	N	Тип	Функция
NF_RnB	PXD[31]	134	I/O	Вход готовности флэш-памяти
NF_CS[2]	PXD[30]	135	I/O	Выход. Выбор банка 2
NF_CS[1]	PXD[29]	136	I/O	Выход. Выбор банка 1

NF_CS[0]	PXD[28]	138	I/O	Выход. Выбор банка 0
NF_CLE	PXD[27]	139	I/O	Выход. Строб команды
NF_ALE	PXD[26]	140	I/O	Выход. Строб адреса
NF_RE	PXD[25]	141	I/O	Выход. Строб чтения
NF_WE	PXD[24]	142	I/O	Выход. Строб записи
NF_D[7]	PXD[23]	143	I/O	Шина данных. Бит 7
NF_D[6]	PXD[22]	144	I/O	Шина данных. Бит 6
NF_D[5]	PXD[21]	145	I/O	Шина данных. Бит 5
NF_D[4]	PXD[20]	146	I/O	Шина данных. Бит 4
NF_D[3]	PXD[19]	147	I/O	Шина данных. Бит 3
NF_D[2]	PXD[18]	149	I/O	Шина данных. Бит 2
NF_D[1]	PXD[17]	150	I/O	Шина данных. Бит 1
NF_D[0]	PXD[16]	151	I/O	Шина данных. Бит 0

## 16.1 Регистры управления контроллером

Поведение контроллера, параметры интерфейса обмена определяются посредством регистров конфигурации контроллера.

Ниже приведен список всех регистров контроллера (Таблица 98). Напомним, что регистры контроллера доступны со стороны шины периферийных устройств и имеют базовый адрес 0x80000240.

**Таблица 98 – Регистры контроллера флэш-памяти**

Смещение	Имя	Описание	Сброс
0	IO_CFG	Регистр конфигурации временных параметров	0x00a3ffff
1	WCT_CFG	Регистр конфигурации времени ожидания	0x000000ff
2	NAND_CFG	Регистр конфигурации протокола обмена	0x0fff035b
3	WR_CFG	Регистр команд записи	0x00100080
4	RD_CFG	Регистр команд чтения	0x00003000
5-7	-	Не используются	
8	CR	Регистр управления	
9	SR	Регистр состояния	
10	AR	Регистр начального адреса обмена	
11	CNTR	Счетчик количества передаваемых слов	
12	DR	Буфер данных.	
13	ERR12	Регистр номеров ошибок 1 и 2	
14	ERR34	Регистр номеров ошибок 3 и 4	
15	SP_BUF	Регистр номера ошибочной подстраницы	

### 16.1.1 IO\_CFG – регистр конфигурации временных параметров

Данный регистр позволяет определить основные временные параметры протокола обмена. Ниже изображены основные сигналы интерфейса обмена и основные временные параметры (Рисунок 52). Длительность стробов выражается количеством тактов частоты синхронизации, на которой работает контроллер. В качестве базовой частоты используется частота периферийной шины.

**Таблица 99 – Регистр IO\_CFG**

Бит	Имя	Описание	Сброс
2:0	CSCA	Число периодов(+1) CLK с момента активизации NF_CS [2:0] до момента активизации сигналов nRE или nWE	111

7:3	CA	Число периодов (+1) активности сигнала nRE во время чтения памяти	11111
10:8	BWD	Число периодов (+1) активности nWE во время записи.	111
14:11	BRT	Число периодов (+1) от момента снятия одного NF_CSx до момента активизации NF_CSx другого банка, с целью избежать конфликт на шине данных	1111
17:15	BHT	Число периодов (+1) CLK в течение которых NF_CSx находится в неактивном состоянии между последовательными выборками. А также число периодов (+1) после снятия nRE или nWE до следующего активного сигнала.	111
18	MDT	Тип интерфейса. Используется совместно с VGA битом	0
20:19	-	Не используется. Значение безразлично	00
21	VGA	Тип интерфейса. Имеет значение при установленном MDT	1
22	-	Не используется. Значение безразлично	
26:23	CSKPL	Управление линиями NF_CSx между доступами к флэш-памяти 0 – линии NF_CSx становятся неактивными между выборками 1 – линии NF_CSx активны (=0) между выборками	0001
31:27	-	Не используется. Значение безразлично	

Отметим функции некоторых бит, значение которых может быть недостаточно понятно из краткого описания в таблице.

Биты MDT и VGA позволяют управлять функцией линий интерфейса: ALE, CLE, nRE (Таблица 100).

**Таблица 100 – Дополнительные функции интерфейса**

MDT	VGA	ALE	CLE	nRE
0	-	ALE	CLE	nRE
1	0	BA[3]	BA[2]	nRE
1	1	nWRITE	BA[2]	STB

Отметим, что строб  $STB = \sim(nRE \ \& \ nWE)$ . Таким образом, можно запрограммировать обмен с устройствами, которые имеют более простой интерфейс обмена. Данная функция работает только в случае, когда фазы передачи команды и адреса выключены.

Биты CSKPL необходимо использовать в следующей ситуации. При чтении или записи выборка кристалла (NF\_CS) активна только на период одного обмена. Поскольку в операциях с памятью всегда используется последовательный режим обмена, то в некоторых микросхемах флэш-памяти снятие выборки после очередного последовательного чтения вызовет останов наращивания адреса внутри флэш-памяти и возврат ее внутренней машины состояния в исходное состояние. Чтобы этого не произошло необходимо установить бит CSKPL = 1 для соответствующего NF\_CS. В этом случае выборка микросхемы памяти будет активна постоянно, начиная с первого обращения.

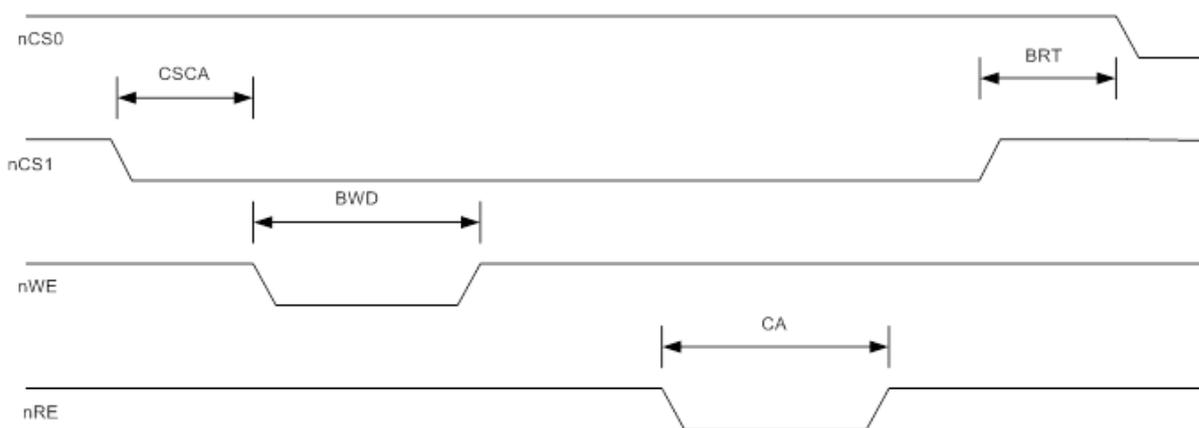


Рисунок 52 – Основные временные параметры обмена

### 16.1.2 WCT\_CFG – регистр конфигурации времени ожидания

Регистр позволяет управлять анализом сигнала готовности внешней памяти и программировать максимальное время ожидания.

Таблица 101 – Регистр WCT\_CFG

Бит	Имя	Описание	Сброс
3:0	ENWT	Разрешение анализа входа RnB во время обмена с одним из банков NF_CS[3:0]. 0 – RnB не анализируется 1 – RnB используется для анализа во время цикла обмена	1111
7:4	CCSE	Управление активностью сигнала выборки банка Эти биты позволяют снять выборку банка во время RnB = 0. Биты регистра 7:4 соответствуют NF_CS[3:0]. 0 – установить неактивный уровень если RnB = 0 1 – разрешить активный уровень на NF_CSx во время RnB=0	1111
30:21	WTOC	Длительность периода “time-out” Эти биты определяют длительность периода, используемого для контроля ситуации “time-out”. Превышение длительности RnB=0 над длительностью периода WTOC*1024 приводит к ситуации “time-out” и установке бита TOEX.	0
31	TOE	Разрешение контроля ситуации “time-out” 0 – запрещено 1 – разрешено	0

Биты ENWT позволяют разрешать для каждого NF\_CS анализ сигнала RnB. При подключении устройств к контроллеру флэш-памяти может оказаться, что некоторые NF\_CS используются для выборки устройств, которые не управляют сигналом RnB. В этом случае необходимо очистить бит ENWT соответствующего NF\_CS.

Функции бит CCSE могут быть использованы в следующем случае. При чтении новой страницы или при записи флэш-памяти вырабатывается сигнал RnB=0 на продолжительное время. Если бит CCSE равен нулю для соответствующего NF\_CS, то в случае RnB = 0 сигнал выборки станет неактивным на период времени, когда RnB равен нулю. Для некоторых типов микросхем это позволяет снизить мощность потребления.

Биты WTOC позволяют запрограммировать интервал времени time-out. При этом будет отслеживаться период равный WTOC\*1024 тактов.

Бит TOE разрешает анализ длительности обмена.

### 16.1.3 NAND\_CFG – регистр конфигурации протокола обмена

Позволяет задать основные параметры флэш-памяти и режимы работы.

Таблица 102 – Регистр NAND\_CFG

Бит	Имя	Описание	Сброс
2:0	VOLCOL	Размер страницы 000 – 128 байта 001 – 512 байта 010 – 1024 байта 011 – 2048 байта 100 – 4096 байта 101 – 8192 байта	011
3	ROWBT	Число байт адреса для выборки строки 0 – 2 байта 1 – 3 байта	1
5:4	COMCLW	Число команд для записи 00 – 1 01 – 2 10 – 3 (расширенный режим) 11 – не используется	01
7:6	COMCLR	Число команд для чтения 00 – 1 01 – 2 10 – не используется 11 – не используется	01
9:8	ADRCL	Число адресных циклов 00 – 1 цикл 01 – только выбор столбца 10 – только выбор строки 11 – строка и столбец	11
10	-	Не используется	0
11	-	Не используется	0
12	ADRSC	Управление пропуском адресного цикла 0 – передавать адрес во время обмена 1 – пропустить	0
13	DATSC	Управление пропуском цикла данных 0 – передавать данные во время обмена 1 – пропустить	0
14		Не используется.	
15	COMSC	Управление пропуском цикла передачи команд 0 – передавать команды во время обмена 1 – пропустить	0
20:16	RESBF	Время задержки до анализа активности RnB Биты задают длительность (*2) периода от момента передачи последнего байта адреса до момента анализа сигнала RnB.	0x1F
27:21	RESAF	Время задержки после завершения активности RnB Биты задают длительность (*2) периода от момента, когда сигнал RnB стал неактивным до момента чтения или записи данных.	0x7F
31:28	ADDRH	Биты адреса расширения	0000

Биты VOLCOL определяют размер страницы флэш-памяти.

Бит ROWBT определяет количество байт адреса, которые необходимо передать во флэш-память для выборки страницы. Для задания номера начального байта внутри страницы используется два байта, что позволяет использовать страницы размером до 64 Кбайт.

Биты COMCLW определяют количество команд (байт команд), используемых при записи данных во флэш-память.

Биты COMCLR выполняют аналогичную функцию при чтении.

Биты ADRCL позволяют управлять передачей адреса.

Если бит ADRSC установлен, контроллер пропускает фазу передачи адреса.

Если бит DATSC установлен, контроллер пропускает фазу управления данными.

Бит COMSC управляет пропуском фазы передачи команд.

При чтении новой страницы флэш-память вырабатывает низкий уровень на выходе RnB. Его длительность равна времени считывания новой страницы из накопителя в буфер (флэш-памяти). Флэш-память изменяет значение на выходе RnB после передачи последнего байта адреса с некоторой задержкой.

Биты RESBF позволяют задержать анализ вывода RnB после передачи адреса перед началом чтения. Это позволяет избежать срабатывания неверного чтения.

Биты RESAF позволяют запрограммировать время готовности памяти после того как на линии RnB установится высокий уровень.

Биты ADDRH позволяют подключить к одному NF\_CS микросхемы памяти объемом больше 4Г байт.

В одном из режимов старта системы предусмотрен старт с начальной загрузкой из флэш-памяти. После сброса контроллер настроен на работу с микросхемой, у которой страница памяти равна 2К байт. Для выбора строки будет использовано три байта адреса, а для выбора столбца два байта адреса. При передаче адреса будут переданы все 5 байт адреса и две команды чтения.

#### **16.1.4 WR\_CFG и RD\_CFG – регистры конфигурации**

Регистры используются для задания значения команд, используемых при чтении или записи флэш-памяти.

**Таблица 103 – Регистр WR\_CFG**

<b>Бит</b>	<b>Имя</b>	<b>Описание</b>	<b>Сброс</b>
7:0	COMW1	первая команда записи	0x80
15:8	COMW2	вторая команда записи	0x00
23:16	COMW3	третья команда записи	0x10
31:24	-		

**Таблица 104 – Регистр RD\_CFG**

<b>Бит</b>	<b>Имя</b>	<b>Описание</b>	<b>Сброс</b>
7:0	COMR1	первая команда чтения	0x00
15:8	COMR2	вторая команда чтения	0x30
23:16	COMR3	третья команда чтения	0x00
31:24	-		

### 16.1.5 CR – регистр управления

Назначение разрядов регистра приведено ниже (Таблица 105). В данном регистре программируются дополнительные параметры обмена.

**Таблица 105 – Регистр CR**

Бит	Название	Функция
0	EN	Разрешение работы (1)
1	RW	Чтение (0) или запись (1)
2	-	
3	SQE	Разрешение последовательного доступа(когда 0).
4	RIM	Разрешение прерывания при чтении (1-разрешение)
5	TIM	Разрешение прерывания при записи (1-разрешение)
6	CIM	Разрешение прерывания при достижении счетчика значения 0 (1-разрешение)
7	-	
9:8	SZ	Размер передаваемых данных 00 – 4 байта, 10 – два байта, 11 – байт
10	-	
11	DBSZ	Размер шины данных FIFO контроллера 0 – 32 бита 1 – 128 бит
12	ERI_EN	Разрешение прерывания в случае обнаружения ошибочной ситуации в работе контроллера, т.е. когда обнаруживается ситуация тайм-аут.
13	RnBI_EN	Разрешение прерывания в случае обнаружения положительного фронта на входе RnB, т.е. когда флаг RnB_Фрегистра состояния равен 1.
14	EIM	1 – разрешение прерывания при обнаружении ошибки ECC
15	ECC_ON	1 – включение режима ECC

После сброса значение регистра управления равно 0x0001, т.е. контроллер включен, настроен на чтение слов с размерностью шины данных в 32 бита.

Далее будут описаны типовые процедуры чтения и записи данных. Они состоят из передачи управляющих команд, адреса и блока данных. Вся эта процедура выполняется, когда бит SQE равен нулю. Если установить бит SQE в единицу, указанный протокол будет выполняться для каждого слова данных, т.е. после передачи команды, адреса и чтения 4-х байт данных (биты SZ равны 00), перед чтением следующих 4-х байт будет повторно выполнена передача команд и адреса. Данный режим очень медленный и не рекомендуется для применения.

Биты разрешения прерывания RIM, TIM, CIM могут быть использованы для организации работы контроллера вместе с процессором. Например, если выполняется чтение данных, разрешение прерывания RIM будет сигнализировать процессору о том, что во внутреннем буфере имеются данные для чтения. Если выполняется запись данных, разрешение прерывания TIM будет сигнализировать процессору о том, что во внутреннем буфере имеется место для записи данных. Прерывание CIM сигнализирует процессору о завершении обмена, т.е. во внутренний буфер или из внутреннего буфера были передано количество слов данных указанное в счетчике CNTR.

Бит RW управляет направлением передачи данных. Значение 0 определяет чтение данных из внешней памяти во внутренний буфер, а значение 1 задает передачу данных из внутреннего буфера во внешнюю память. Отметим, что чтение данных из внешней памяти инициируется только, если есть свободное место во внутреннем буфере, а запись во внешнюю память инициируется только когда есть данные во внутреннем буфере.

Бит EN разрешает работу контроллера. Если в данный бит записать ноль, произойдет очистка внутренних указателей внутреннего буфера и перевод машины состояний в исходное положение.

Биты SZ задают размер слова передаваемых данных. Размер 1 или 2 байта должен использоваться только для спецопераций.

Биты DBSZ задают размер шины данных FIFO. При смене размера шины необходимо выключить контроллер NAND флэш-памяти.

### **16.1.6 DR – регистр данных**

Контроллер имеет FIFO данных объемом восемь 32-разрядных слов. Оно выполняет функцию буфера между системой и внешней флэш-памятью. При чтении данных из флэш-памяти, данные записываются в FIFO. При записи – данные считываются из FIFO и передаются в флэш-память. Можно установить размер шины данных FIFO. Если бит DBSZ равен нулю, размер FIFO составит 8x32 бита, а если DBSZ равен 1, размер FIFO – 2x128. В последнем случае процессор или DMA должны производить обмен квадрословами. Признаки готовности данных также формируются с учетом размерности шины данных.

Рассмотрим поведение буфера данных в зависимости от размерности шины данных FIFO.

#### **1. Шина данных 32 бита:**

В этом случае буфер имеет размер 8 слов по 32 бита. Имеются указатели на запись и чтение каждый разрядностью 3 бита. Запись в буфер увеличивает указатель записи на 1. Чтение из буфера уменьшает указатель чтения на 1.

Признак готовности буфера принимать данные – TFS активен (равен 1) в случаях, когда буфер пуст, или когда буфер не заполнен полностью, т.е.  $TFS==1$  означает что в буфер можно записать одно слово данных.

Признак RFS (для случая, когда буфер настроен на чтение) активен (равен 1) в случаях, когда имеются данные для чтения, т.е. когда буфер не пуст. Это означает, что из буфера можно прочесть одно слово данных.

#### **2. Шина данных 128 бит:**

В этом случае буфер имеет разный вид со стороны процессора и со стороны контроллера флэш-памяти. Для процессора буфер имеет размер 2 слова по 128 бит, а для контроллера по-прежнему 8 слов по 32 бита. Указатели на запись и чтение изменяются в зависимости от направления передачи:

##### **– Запись**

Запись в буфер увеличивает указатель записи на 4, т.к. в буфер помещаются сразу 4 слова данных. Чтение из буфера уменьшает указатель чтения на 1. Признак готовности буфера принимать данные TFS активен (равен 1) в случаях, когда буфер пуст или, когда буфер не заполнен полностью, т.е.  $TFS==1$  означает, что в буфер можно записать одно квадрослово данных.

##### **– Чтение**

Запись в буфер со стороны флэш-памяти увеличивает указатель записи на 1, т.к. в буфер помещается 1 слово данных. Чтение из буфера уменьшает указатель чтения на 4, т.к. считывается сразу 4 слова. Признак RFS активен (равен 1) в случаях, когда имеется хотя бы одно квадрослово

данных для чтения, т.е. когда из памяти прочитаны как минимум 4 слова. Это означает, что из буфера можно прочитать одно квадрослово данных.

Бит разрядности шины данных имеет отношение только к буферу данных. Доступ к регистрам управления осуществляется только словами по 32 бита. При смене разрядности шины данных буфера **обязательно** нужно выполнить выключение контроллера (CR[0]=0). Это приведет к обнулению указателей чтения и записи и таким образом гарантирует их дальнейшую корректную работу.

Для операций с флэш-памятью, требующих чтения менее, чем 4-х слов данных необходимо использовать только 32-разрядный режим шины данных буфера.

### **16.1.7 AR – регистр адреса**

Перед началом обмена необходимо задать адрес, начиная с которого будет производиться чтение или запись. В последующем этот адрес будет автоматически увеличиваться на 1. При записи значения в регистр адреса необходимо помнить, что данный адрес есть адрес слова. Внутри контроллера адрес расширяется AF[33:0]={AR[31:0],2'b00} двумя нулевыми битами для формирования указателя на байт. Биты AF[33:32] используются для выбора микросхемы памяти, а биты AF[31:0] прямо адресуют память. Это позволяет адресовать 4 Гбайт памяти. Внутри контроллера биты AF[31:0] дополняются битами расширения адреса AX[35:0]=={ADDRH[3:0],AF[31:0]}. Биты расширения имеют значение, если объем подключенной микросхемы памяти более 4 Гбайт.

Внутри контроллера исполнительный адрес AX делится на адрес столбца и адрес строки. Значимым параметром является адрес столбца (column). Количество бит выделяемое для адреса столбца зависит от размера страницы флэш-памяти. Например, для случая, когда страница равна 2К байт в качестве адреса столбца будут использованы биты AX[10:0]. Биты AX[35:11] будут адресовать строку.

### **16.1.8 CNTR – счетчик количества слов**

Задаёт количество слов, которое будет прочитано из флэш-памяти или записано во флэш-память. Счетчик имеет разрядность 24 бита. При этом длина слова может быть 1, 2 или 4 байта.

### **16.1.9 SR – регистр состояния**

Отражает значение флагов запросов прерываний. Биты регистра описаны ниже (Таблица 106).

**Таблица 106 – Регистр SR**

<b>Бит</b>	<b>Название</b>	<b>Функция</b>
0	EMPTY	FIFO данных пусто (1)
1	FULL	FIFO данных заполнено полностью (1)
2	TFS	Запрос от FIFO данных на запись новых данных процессором или контроллером прямого доступа. FIFO наполовину пусто (1).
3	RFS	Запрос от FIFO данных на чтение новых данных процессором или контроллером прямого доступа. FIFO наполовину заполнено (1).
4	IRQ	Общий запрос прерывания от контроллера. Более детально причина запроса прерывания может быть установлена согласно состоянию бит

		20:16 регистра. 0 – нет запроса 1 – запрос на прерывание
7:5	-	-
11:8	TOEX	Флаги состояния Time-Out. Биты отражают факт обнаружения ситуации “time-out” в процессе обмена с соответствующим NF_CSx. 0 – нет ошибки 1 – обнаружена ситуация “time-out” Биты очищаются программно, посредством записи в них 1.
12	-	
13	RnB_F	Флаг изменения входа RnB из низкого уровня «0» в высокий «1» 0 – не было изменения 1 – был переход линии RnB из 0 в 1 Бит очищается программно, посредством записи в него 1
15:14	-	
16	RFS_I	Запрос на прерывание когда RFS==1 и бит RIM==1. Запрос может быть очищен посредством чтения буфера данных.
17	TFS_I	Запрос на прерывание когда TFS==1 и бит TIM==1. Запрос может быть очищен посредством записи в буфер данных.
18	CNTZ_I	Запрос на прерывание когда значение счетчика становится равным нулю и бит CIM==1. Запрос может быть очищен посредством записи нового значения счетчика.
19	TO_I	Запрос на прерывание когда TOEX==1 и бит ERI_EN ==1. Запрос может быть очищен посредством записи 1 в бит TOEX.
20	RnBF_I	Запрос на прерывание когда RnB_F ==1 и бит RnBI_EN ==1. Запрос может быть очищен посредством записи 1 в бит RnB_F.
30:21	-	-
31	RnB	Вход готовности NAND флэш-памяти. Отражает состояние линии RnB.

Контроллер интерфейса содержит внутренний счетчик, который позволяет отследить ситуацию, когда обмен с памятью превышает допустимый временной интервал (ситуация time-out). Это служит защитой от блокировки работы процессора, выйти из которой можно только сбросив всю систему.

Если ситуация time-out произошла при обращении к флэш-памяти, подключенной к некоторому NF\_CS выводу, соответствующий бит TOEX в регистре состояния будет установлен и цикл обмена завершится некорректно, т.е. считанные данные будут неверными или запись будет осуществлена неверно. Подобная ситуация может произойти если длительность низкого уровня на входе RnB превышает период time-out. Процесс аварийного завершения обмена может вызвать прерывание работы процессора, если соответствующее прерывание разрешено.

Сбросить биты TOEX можно только записав в них значение 1. Запись 0 в соответствующие биты не вызывает их изменения.

Бит RnB отражает состояние соответствующего выхода готовности микросхемы флэш-памяти. Бит может быть использован для определения готовности флэш-памяти.

## **16.2 Чтение NAND флэш-памяти**

Чтение флэш-памяти всегда осуществляется в последовательном режиме и представляет собой передачу во внешнюю память команд, байт адреса и затем чтение непосредственно самих данных. Различные варианты чтения отличаются друг от друга количеством передаваемых байт команд и адреса, а также количеством принимаемых байт данных.

При чтении данных вначале происходит передача одного (или двух) байта команды, затем передаются байты адреса, после этого читаются данные. Первую команду, с которой начинается чтение, необходимо записать в поле COMR1 регистра RD\_CFG.

Если размер страницы равен 512 байт, команды для доступа к первой половине страницы и ко второй половине различны (00h – для первой, 01h – для второй). Контроллер отслеживает эту ситуацию сам. Пользователь должен записать в поле COMR1 значение 00h и определить размер страницы как 512 байт. Если при чтении контроллер обнаруживает, что обращение осуществляется ко второй половине страницы, он автоматически устанавливает команду 01h вместо 00h.

После передачи команды контроллер посылает во флэш-память байты адреса. Количество адресных байт определяется в регистре конфигурации и зависит от емкости микросхемы памяти. После этого контроллер читает заданное количество байт данных последовательно и помещает их во внутренний буфер.

Передаваемое значение адреса формируется из регистра адреса AR (биты 29:0) и поля расширения адреса ADDRH(биты 2:0). Поле расширения определяет старшие биты передаваемого адреса.

Контроллер ориентирован на чтение количества байт кратных 4. Однако для выполнения специальных операций (сброс микросхемы, чтение регистра состояния микросхемы) можно выполнять более короткие операции чтения. Для чтения одного байта необходимо в счетчик слов CNTR записать значение 1 и установить размер передаваемых данных в регистре управления CR равным байту. Для чтения 2-х байт в счетчик слов записываем 1, а размер данных устанавливаем равным полуслову.

### **16.3 Запись в NAND флэш-память**

Операция записи во многом аналогична операции чтения только направление передачи данных противоположное, т.е. из внутреннего буфера во внешнюю микросхему памяти. Команды, используемые при записи во флэш-память, должны быть записаны в поля регистра конфигурации WR\_CFG. В типичном случае протокол записи данных в память состоит из передачи первого командного байта, передачи байт адреса, записи данных и передачи второго байта команды, инициирующего процесс записи внутри флэш-памяти.

Необходимо отметить, что во время программирования текущей страницы во внутреннем накопителе, флэш-память становится недоступной. Это занимает довольно продолжительное время. При выполнении операций контроллер анализирует готовность памяти и в случае обращения процессора за новыми данными в момент программирования предыдущих данных, будет осуществлена остановка работы контроллера до момента готовности флэш-памяти.

Для записи данных необходимо определить две команды: для инициализации процесса записи и для его окончания. Эти две команды записываются в поля COMW1 и COMW3 регистра конфигурации. Вначале посылается первая команда, затем байты адреса, байты данных. Во время пересылки данных во внешнюю память контроллер анализирует состояние счетчика CNTR(после пересылки каждого слова значение счетчика уменьшается на 1) и в случае равенства счетчика значению 1, контроллер отсылает последнее слово данных и за ним следует вторая команда записи. После приема второй команды записи микросхема памяти начинает процесс перезаписи принятых данных во внутренний накопитель. Флэш-память тратит значительное время для перезаписи внутреннего буфера в накопитель. На все это время память становится недоступной. Имеется возможность анализировать сигнал готовности посредством чтения регистра состояния.

## 16.4 Типичные процедуры работы с NAND флэш-памятью фирмы Samsung

### 16.4.1 Сброс машины состояния флэш-памяти

- **IO\_CFG** = К, устанавливаем временные параметры соответствующие подключенной микросхеме. Если тип микросхемы неизвестен, используем максимальные значения временных параметров (т.е. худший случай)
- **NAND\_CFG** = 0x3000, устанавливаем пропуск фаз передачи адреса и чтения данных, число команд чтения равно 1.
- **RD\_CFG** = 0xFF, определяем первую команду чтения (COMR1).
- **CR** = 0x0301. Устанавливаем чтение байта.
- **CNTR** = 1, иницилируем процесс чтения.
- происходит фиктивное чтение байта из флэш-памяти. Поскольку фазы адреса и данных будут пропущены, переданы будут только команды **FFh** во флэш-память. Тем не менее, нужно помнить, что байт данных (неизвестное значение) все равно будет загружен во внутренний буфер, и этот байт нужно вычитать, чтобы привести указатели в исходное положение.

### 16.4.2 Чтение ID

Согласно спецификации, для чтения ID необходимо передать команду **90h**, затем один байт адреса (значение 0) и затем прочитать два байта ID. Как это можно реализовать в данном ПЦОС?

- здесь и далее предполагаем, что **IO\_CFG** был определен ранее
- **RD\_CFG** = 0x90, определяем первую команду чтения (COMR1).
- **NAND\_CFG** = 0, устанавливаем передачу только одного байта адреса, число команд чтения равно 1.
- **CR** = 0x0001. Устанавливаем чтение слова.
- **AR** = 0, адрес равен нулю
- **CNTR** = 1, иницилируем процесс чтения.
- выполняется чтение слова по адресу 0. Прочитанные четыре байта и есть ID.

### 16.4.3 Чтение регистра состояния

Согласно спецификации, для чтения состояния необходимо передать команду **70h**, а затем прочитать байт состояния, т.е.:

- здесь и далее предполагаем, что **IO\_CFG** был определен ранее
- **RD\_CFG** = 0x70, определяем первую команду чтения (COMR1).
- **NAND\_CFG** = 0x1000, устанавливаем пропуск адреса, число команд чтения равно 1.
- **CR** = 0x0301. Устанавливаем чтение байта.
- **CNTR** = 1, иницилируем процесс чтения.
- Прочитанный байт и есть регистр состояния. При этом необходимо помнить, что перед чтением данных из внутреннего буфера необходимо убедиться в их наличии. Для этого необходимо сканировать регистр состояния SR и ожидать установки бита RFS в единицу. Это будут сигнализировать о том, что операция завершилась и данные готовы. При чтении регистра состояния внешней памяти необходимо помнить, что если

память занята (например, производит стирание блока), то она может притормозить контроллер посредством сигнала RnB. В этом случае перед чтением регистра состояния лучше выключить анализ RnB для выбранного банка памяти.

#### 16.4.4 Очистка (erase) блока флэш-памяти

Согласно спецификации, для стирания блока флэш-памяти необходимо передать команду **60h**, передать адрес блока, пропустить запись данных и в заключение передать команду **d0h**. Таким образом, последовательность действий может быть следующей:

- **RD\_CFG** = 0xd060, определяем первую (COMR1) и вторую (COMR2) команды чтения.
- **NAND\_CFG** = 0x2248, устанавливаем пропуск данных, число команд чтения равно 2, передачу только адреса для выбора блока.
- **AR** = address, записываем в регистр адреса значение соответствующее номеру стираемого блока. Здесь необходимо рассчитать какие именно биты адреса регистра AR используются для указания номера блока. Если размер страницы указан как 2 Кбайт, то биты AR[8:0] будут использоваться для адресации байта внутри страницы (плюс дополнительно два нулевых бита). Биты AR[29:9] будут указывать на номер страницы. Стираемый блок может состоять из нескольких страниц. Если такое число страниц равно, например, 64, то передаваемые биты AR[14:9] безразличны. Учитываются только биты AR[29:15].
- **CR** = 0x0301. Устанавливаем чтение байта.
- **CNTR** = 1, иницируем процесс чтения.
- выполняется фиктивное чтение байта. Это приводит к передаче во флэш-память требуемой последовательности байт. После этого можно анализировать флаг готовности флэш-памяти путем чтения регистра состояния памяти или путем анализа флага RnB регистра состояния SR. Также не забываем о прочитанном байте в буфере.

#### 16.4.5 Запись во флэш-память в последовательном режиме

Согласно спецификации, для записи данных необходимо передать команду **80h**, передать адрес, записать данные последовательно в текущую страницу, после этого необходимо передать команду **10h**, которая иницирует процесс программирования. Таким образом:

- **WR\_CFG** = 0x100080, определяем первую команду записи (COMW1=80h), вторая команда записи не используется, третья команда записи (COMW3=10h).
- **NAND\_CFG** = 0x035B, устанавливаем передачу полного адреса, число команд записи равно 2, размер страницы 2 Кбайт.
- **AR** = address, записываем в регистр адреса значение начального адреса слова(!) начиная с которого мы хотим произвести запись блока данных.
- **CR** = 0x0043. Устанавливаем операцию записи слова размером 4 байта и прерывание по достижению счетчика значения ноль.
- **CNTR** = K, иницируем процесс записи K\*4 байт данных.
- выполняем запись последовательного блока данных во внутренний буфер анализируя при этом его доступность для записи. Как только мы записали

последнее слово в буфер, переходим к ожиданию прерывания о завершении операции записи.

#### 16.4.6 Чтение из флэш-памяти в последовательном режиме

Согласно спецификации, для чтения данных необходимо передать команду **00h(01h, 50h)**, передать адрес, прочитать данные последовательно. Таким образом:

- **RD\_CFG** = 0x3000, определяем команды чтения (COMR1=00h, COMR2=30h).
- **NAND\_CFG** = 0x035B, устанавливаем передачу полного адреса, число команд записи равно 2, размер страницы 2 Кбайт.
- **AR** = address, записываем в регистр адреса значение начального адреса слова(!) начиная с которого мы хотим произвести чтение блока данных.
- **CR** = 0x0001. Устанавливаем операцию чтения слова размером 4.
- **CNTR** = K, инициуем процесс чтения K\*4 байт данных.
- переходим к анализу флага состояния RFS буфера данных. Как только он устанавливается в 1, производим чтение данных. После прочтения всех данных завершаем операцию.

#### 16.5 Рекомендации по организации работы с прерываниями

Для корректной работы NAND контроллера с контроллером DMA нужно будет строго соблюдать последовательность действий. При этом эта последовательность будет разной при чтении и при записи. Исходное состояние – все выключено.

##### При чтении:

- программируем канал DMA. После включения он ждет импульса от NAND контроллера.
- программируем NAND контроллер и включаем обмен.
- при чтении данных из памяти в FIFO генерируется запрос к контроллеру DMA (или прерывание процессору) о том, что можно выполнить дальнейшую пересылку.
- Контроллер DMA пересылает данные.
- после чтения последнего слова NAND-контроллер может сгенерировать прерывание для процессора. Однако это не очень разумно, т.к. данные еще не переместились в память. Поэтому при чтении лучше использовать прерывание от канала DMA.
- запрос прерывания от контроллера DMA говорит о том, что все операции завершены.

##### При записи:

- программируем канал DMA. После включения он ждет импульса от NAND-контроллера.
- программируем NAND-контроллер и включаем обмен, сразу идет запрос к контроллеру DMA.
- контроллер DMA пересылает данные в FIFO и они передаются в память.
- после пересылки последнего слова контроллер DMA может генерировать прерывание. Но при записи лучше использовать прерывание от NAND. Оно будет говорить о том, что данные уже ушли в память.

## 17 Контроллер LCD-панели

Возможно подключение LCD-панелей (тип TFT) к процессору. Параметры контроллера позволяют поддерживать панели различных разрешений.

Структура контроллера LCD-панели представлена ниже (Рисунок 53).



Рисунок 53 – Структура контроллера

Контроллер осуществляет прием входной информации, ее преобразование и выдачу на внешнюю LCD-панель в соответствии с заданным режимом работы. Задание режима работы контроллера осуществляется посредством программирования его регистров. Регистры контроллера LCD-панели подключены к пользовательской периферийной шине и имеют базовый адрес 0x8000\_0160. Перечень регистров приведен ниже (Таблица 107).

Таблица 107 – Регистры контроллера

Номер	Название	Функция
0	CTRL	Регистр управления
1	STATUS	Регистр состояния
2	HTIM	Fpline (старт, стоп)
3	VTIM	Fpframe (старт, стоп)
4	HVLEN	Размер экрана
5	HDxTIM	Горизонтальная выдача данных(старт, стоп). Дополнительное окно
6	VDxTIM	Вертикальная выдача данных (старт, стоп)). Дополнительное окно
7	Vsize	Размер видеобuffers
8	FON	Данные для заполнения фоновой области
9	PXDV	Делитель пиксель блока
10	HDTIM	Горизонтальная выдача данных (старт, стоп). Основное окно.
11	VDTIM	Вертикальная выдача данных (старт, стоп). Основное окно.
12	Panel_CFG	Регистр конфигурации панели
13	Pwm_CR	Регистр управления ШИМ
14	SLP_period	Регистр задания интервала сна
15	-	
16	Tim_GP0	Управление сигналом GPIO_0
17	Tim_GP1	Управление сигналом GPIO_1
18	Tim_GP2	Управление сигналом GPIO_2
19	Tim_GP3	Управление сигналом GPIO_3

Управление внешней LCD панелью осуществляется посредством набора внешних выходов. Внешние выводы контроллера состоят из 26-разрядного порта

общего назначения и выхода синхронизации PIX\_CLK(или FPSHIFT). Назначение выводов порта общего назначения приведено ниже (Таблица 108).

**Таблица 108 – Внешние выводы управления панелью**

<b>Разряды порта</b>	<b>название</b>	<b>функция</b>
17:0	fpdat	Шина данных пикселей
18	Gpio_0	Дополнительное управление панелью
19	Gpio_1	Дополнительное управление панелью
20	Gpio_2	Дополнительное управление панелью
21	Gpio_3	Дополнительное управление панелью
22	pwm	Выход ШИМ
23	drdy	Готовность данных
24	fpframe	Импульс начала фрейма
25	fpline	Импульс начала линии
26	fpshift	Пиксель-клок

Входной буфер имеет входную шину данных разрядностью 128 бит для загрузки данных. Емкость буфера составляет пять 128-разрядных слов. Входной буфер формирует запрос к контроллеру ДМА на прием данных, если он не заполнен. Из входного буфера данные поступают на преобразователь порциями по 32 бита. Модуль преобразователя осуществляет формирование информации о RGB-пикселе на основании полученной информации и заданных блоком управления параметрах преобразования. На выходе преобразователя всегда данные RGB пикселя разрядностью 18 бит. Каждая цветовая компонента имеет разрядность 6 бит. Эти данные поступают в выходной буфер. Емкость выходного буфера составляет 256 пикселей. При формировании временной диаграммы управления панелью, данные из выходного буфера в нужный момент времени выдаются на внешнюю панель. В задачу преобразователя входит постоянная подкачка данных в выходной буфер по мере его освобождения. Использование выходного буфера позволяет осуществить равномерную выдачу информации на панель, даже в случаях, когда ДМА контроллер может быть заблокирован на некоторое время более высокоприоритетными обменами.

Регистры контроллера позволяют настроить контроллер на работу с различными типами LCD панелей. При этом возможно задание, как дополнительных внешних управляющих линий, так и настройка контроллера для работы с выбранным разрешением экрана. Регистры управления позволяют задать количество вертикальных линий панели, а также количество пикселей в линии. Дополнительно задается область выдачи данных, т.к. количество линий и пикселей в линии может превышать количество пикселей задействованных в отображении информации. В зависимости от требований панели, контроллер может выполнять как непрерывную выдачу данных (автоматический переход на начало повторной выдачи картинки после завершения текущей выдачи), так и выдачу данных по запросу. В последнем случае контроллер после завершения выдачи картинки переходит в спящий режим и начинает новую выдачу по истечению запрограммированного интервала времени или по требованию процессора. Данный режим может использоваться, только если панель (внешнее устройство-приемник) допускает останов пиксель блока на некоторое время.

В принципе, контроллер LCD панели можно также рассматривать как параллельный синхронный 16-разрядный порт для выдачи информации, с настраиваемыми параметрами выдачи данных и генерации сигналов управления.

Ниже приведена типовая временная диаграмма работы контроллера (Рисунок 54). Задача контроллера сводится к выдаче на внешние контакты одного фрейма (картины) изображения. Начало картины всегда начинается с выдачи активного уровня FFRAME (на рисунке активный низкий уровень). Изображение состоит из некоторого множества линий. Начало каждой линии сопровождается активным уровнем FPLINE (на рисунке активный низкий уровень). Информация о пикселях текущей линии выдается только при активном уровне на линии готовности данных DRDY (на рисунке активный высокий уровень). Значение пикселя на шине данных FPDAT выдается относительно спадающего фронта (из высокого в низкий) синхросигнала FPSHIFT. Нарастающим фронтом синхросигнала данные должны приниматься в устройстве приемнике информации.

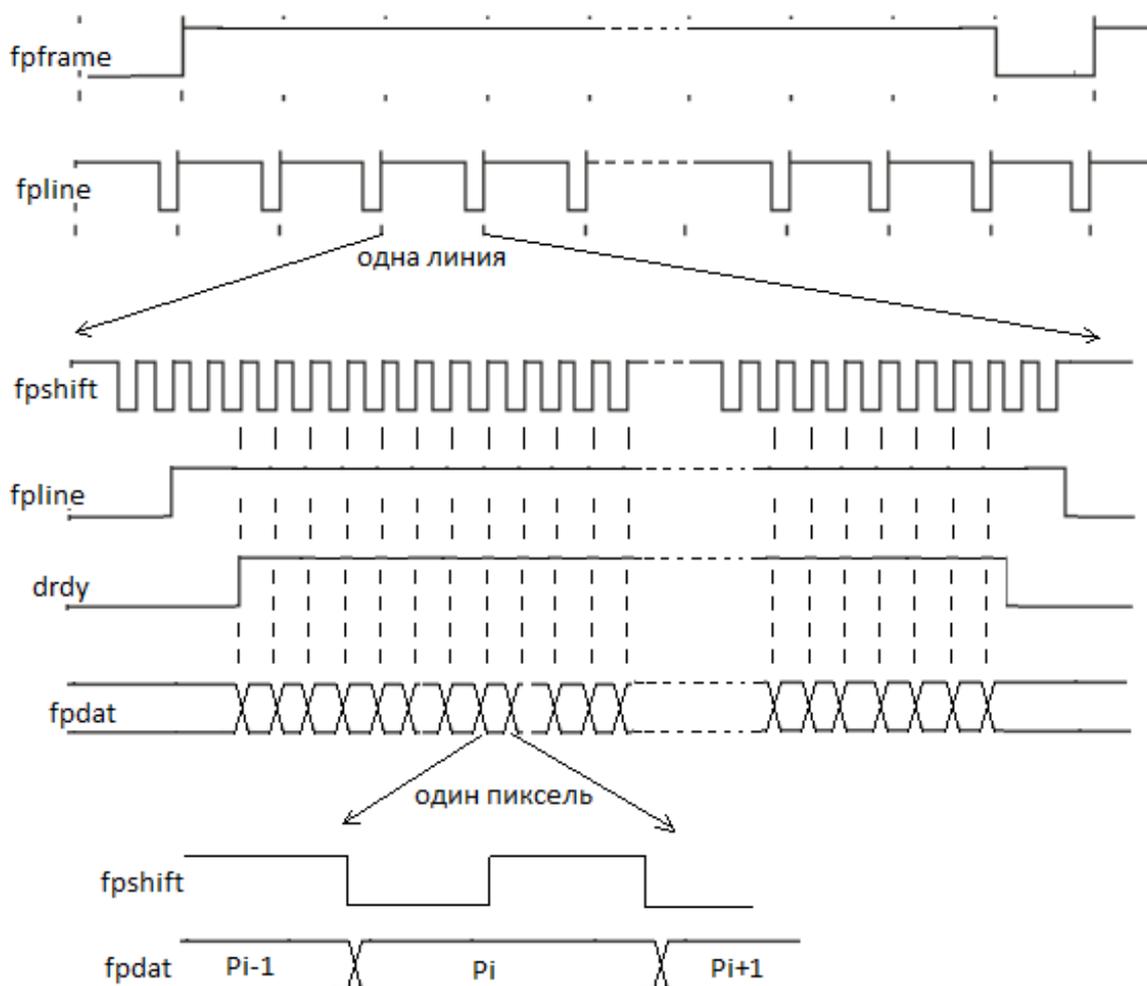


Рисунок 54 – Временная диаграмма работы контроллера

Регистры управления контроллером предназначены для задания различных временных характеристик обмена. После выдачи одного фрейма контроллер повторяет свои действия, но, возможно, уже с другими пиксельными данными.

## 17.1 Регистр управления CTRL

Назначение разрядов регистра управления приведено ниже (Таблица 109).

**Таблица 109 – Разряды регистра управления CTRL**

Бит	Имя	Функция
0	ven	Разрешение работы контроллера: 0 – выключен 1 – включен
1	vie	Разрешение прерывания после окончания фрейма: 0 – запрещено 1 – разрешено
2	hie	Разрешение прерывания после окончания одной линии: 0 – запрещено 1 – разрешено
3	vbie	Разрешение прерывания после загрузки всего видеобанка данных в выходной буфер
4	slpie	Разрешение прерывания после завершения сна
5	-	
6	-	
8:7	vbl	Количество бит в пикселе. Зависит от выбранного режима (биты CD) Таблица 110
10:9	cd	Количество бит для одного пикселя входной информации: 00 – 8 01 – 16 10 – 24 11 – 32
11	-	
12	hldm	Останов повторной загрузки видеоданных после завершения выгрузки видеобуфера (1 – разрешено)
13	hldv	Останов повторной генерации управляющих сигналов панели после завершения выгрузки видеобуфера: 0 – останов запрещен 1 – останов разрешен
14	-	
15	bl	Активный уровень линии готовности данных DRDY (0 – высокий, 1 – низкий)
16	-	
17	vbgr	1 – BGR режим, 0 – RGB режим
18	-	
19	-	
20	Slp_mode	1 – спящий режим после выгрузки видеобуфера
21	Slp_pclk	Стоп делитель пиксель клона в спящем режиме(1)
22	Slp_pxen	Стоп разрешения работы пиксель клона в спящем режиме
23	Slp_hold	Останов повторной генерации управляющих сигналов панели после завершения выгрузки видеобуфера(1 – разрешено)
24	Slp_clrf	Автоматический сброс флагов после задержки спящего режима если 1. В случае 0 – ожидание программного обслуживания интерфейса
25	-	
26	-	
27	-	
28	W2W_en	Разрешение использования регистров дополнительной активной области: 0 – запрещено 1 – разрешено
29	PXP_en	Разрешение останова пиксельклока в случае если в выходном FIFO нет данных:

		0 – запрещено. Будет генерироваться ошибка 1 – разрешено. Клок продолжит тактирование когда будут данные
--	--	---

Бит VEN включает контроллер. Этот бит должен быть установлен в единицу, только когда все параметры контроллера определены.

Биты VIE и HIE позволяют задать формирование запроса прерывания после окончания выдачи всей картинки (VIE) или после окончания очередной одной горизонтальной линии (HIE). Данные прерывание имеют скорее отладочную функцию, т.к. их сложно использовать при нормальном функционировании контроллера.

Бит VBIE позволяет сформировать запрос прерывания после завершения загрузки последнего 32-разрядного слова фрейма из входного буфера в преобразователь. Генерация данного прерывания означает завершение чтения всех данных видеофрейма.

Бит SLPIE позволяет сформировать запрос прерывания к процессору после истечения периода сна контроллера.

Биты CD задают количество бит во входном слове, которые используются для кодирования информации о пикселе. Значения 0, 1, 2 и 3 соответствуют 8, 16, 24 и 32 битам.

В случае использования 32 бит, старшие 8 бит отбрасываются. Использование 32 бит равносильно режиму 24 бит, однако информация о каждом пикселе выровнена на границе слова. Из-за этого впустую теряется один байт. Режим 8 бит может использоваться для режима черно-белого изображения, т.к. в этом случае каждая цветовая RGB компонента равна значению байта. В случае использования 16-бит режима возможны несколько вариантов кодирования. Эти варианты кодируются с использованием битов VBL. Возможные варианты кодирования приведены ниже (Таблица 110). Значение VBL равно нулю (или 1) соответствует стандартному формату RGB565.

**Таблица 110 – Режимы формирования пиксельных данных**

CD	VBL	byte	R	G	B
11	-	-	Din[23:16]	Din[15:8]	Din[7:0]
10	-	-	Din*[23:16]	Din*[15:8]	Din*[7:0]
00	-	0	Din[7:0]	Din[7:0]	Din[7:0]
00	-	1	Din[15:8]	Din[15:8]	Din[15:8]
00	-	2	Din[23:16]	Din[23:16]	Din[23:16]
00	-	3	Din[31:24]	Din[31:24]	Din[31:24]
01	0x	-	{ Din[15:11],000}	{ Din[10:5],00}	{ Din[4:0],000}
01	10	-	{ Din[13:10],0000}	{ Din[8:5],0000}	{ Din[3:0],0000}
01	11	-	{ Din[14:11],0000}	{ Din[9:6],0000}	{ Din[4:1],0000}

Из всех приведенных режимов, отметим режим RGB565, как единственный случай, при котором на внешние контакты информация выдается без потерь входной информации. Данный режим позволяет рассматривать LCD порт как синхронный 16-разрядный порт.

Если используемая LCD панель допускает кратковременную остановку сигнала синхронизации, полезным будет использование бита PXP\_en. Если этот бит установлен, то в случае, когда необходимо выдать новое значение данных и выходной буфер пуст, контроллер остановит сигнал синхронизации панели до момента готовности данных в выходном буфере. Это гарантирует отсутствие сбоев при выдаче информации.

## 17.2 Регистр состояния STATUS

Регистр STATUS отражает текущее состояние запросов прерываний к процессору.

**Таблица 111 – Регистр состояния STATUS**

Бит	Имя	Функция
0	-	
1	luint	Флаг прерывания при ошибке чтения выходного буфера. Если флаг установлен, означает, что при необходимости выдачи данных, буфер был пуст. Сбрасывается только при записи в данный бит значения 1.
2	Sta_sleep	Таймер сна начал отсчет. Устанавливается в момент перехода контроллера в режим сна.
3	Fin_sleep	Таймер сна закончил отсчет. Устанавливается в момент завершения режима сна.
4	vint	Флаг завершения выдачи фрейма.
5	hint	Флаг завершения выдачи линии.
6	vbsint	Флаг завершения чтения данных видеобуфера. Устанавливается при передаче последнего слова данных из входного буфера в преобразователь. Сбрасывается программно или аппаратно.
-		
20	Sleep_exe	Режим сна (если 1). Только чтение.

Флаги регистра устанавливаются аппаратно. Сброс флагов может выполняться программно путем записи в необходимый бит значения 1. Также флаги могут сбрасываться аппаратно в момент выхода контроллера из режима сна. Флаг luint может быть сброшен только программно.

## 17.3 Регистр управления сигналом fpline (HTIM)

Этот регистр осуществляет управление импульсом начала линии и задает смещение импульса относительно начала строки, а также ширину импульса и его активный уровень. Единицей измерения является длительность PX\_CLK сигнала.

**Таблица 112 – Регистр состояния управления сигналом fpline (HTIM)**

Бит	Имя	Функция
9:0	hps	FPLINE начальная позиция (hps+1)
15	hpl	FPLINE активный уровень (0 – высокий, 1 – низкий)
25:16	hpw	FPLINE конечная позиция (hpw+1)

## 17.4 Регистр управления сигналом fframe (VTIM)

Этот регистр осуществляет управление импульсом начала фрейма и задает смещение импульса относительно начала фрейма, а также ширину импульса и его активный уровень. Единицей измерения является одна строка.

**Таблица 113 – Регистр управления сигналом fframe (VTIM)**

Бит	Имя	Функция
9:0	vps	FPFRAME начальная позиция
15	vpl	FPFRAME активный уровень (0 – высокий, 1 – низкий)
25:16	vpw	FPFRAME конечная позиция (vpw+1)

## 17.5 Регистр управления размером экрана (HVLEN)

Этот регистр задает количество пикселей в одной линии и количество линий на экране. Не все пиксели линии могут быть отображаемыми на экране. Также не все линии могут быть отображаемыми на экране. Регистр HVLEN задает общее количество пикселей и линий, которое включает всю информацию (отображаемую и нет).

**Таблица 114 – Регистр управления размером экрана (HVLEN)**

Бит	Имя	Функция
6:0	ht	Размер по горизонтали $((ht + 1) * 8)$
25:16	vt	Размер по вертикали $(vt+1)$

## 17.6 Регистр размера видео буфера (VSIZE)

Регистр задает количество 32-разрядных слов, которые образуют видеобуфер. Контроллер использует это число, чтобы отследить момент окончания загрузки данных и сформировать запрос на прерывание. Момент формирования запроса прерывания соответствует чтению последнего слова видеобуфера из входного FIFO в преобразователь.

**Таблица 115 – Регистр размера видео буфера (VSIZE)**

Бит	Имя	Функция
19:0	VSIZE	Размер видеобуфера в 32-бит словах + 1
31:20	-	

После чтения последнего слова в преобразователь, происходит

- сброс бита 5 регистра управления (не используется).
- установка бита 6 регистра состояния (флаг завершения чтения видеобуфера данных).

Контроллер всегда используется совместно с выбранным для него каналом DMA. В канале DMA также задается число слов для передачи и запрос прерывания от канала DMA может быть более информативным, чем запрос прерывания от LCD контроллера на обслуживание видеобуфера.

Если бит 12 (HLDM) регистра управления установлен, после установки бита 6 регистра состояния, входное FIFO будет постоянно посылать в преобразователь признак того, что оно пусто. При этом входной буфер может формировать запрос к контроллеру DMA и если канал работает, то во входной буфер может быть загружено до пяти квадрослов данных. Данные из входного буфера не будут поступать в преобразователь до тех пор, пока бит 6 регистра состояния не будет сброшен. Бит может быть сброшен программно или аппаратно при выходе из режима сна.

## 17.7 Делитель для сигнала синхронизации панели (PXDV)

LCD контроллер использует для своей работы сигнал синхронизации периферийной шины SOC\_CLK. Данный синхросигнал используется для формирования сигнала синхронизации пикселей PX\_CLK.

$$PX\_CLK = SOC\_CLK / (P\_div + 1)$$

Таблица 116 – Делитель для сигнала синхронизации панели (PXDV)

Бит	Имя	Функция
7:0	P_div	Значение делителя
8	En_div	Включение делителя: 0 – выключен 1 – включен

## 17.8 Управление горизонтальной активной областью панели (HDTIM)

Этот регистр осуществляет управление видимой областью линии и задает смещение данной области относительно начала строки, а также ширину области. Единицей измерения является длительность PX\_CLK сигнала. Как отмечалось ранее, регистр HVLEN задает общее количество пикселей в строке. Счетчик пикселей строки будет изменяться от 0 до Np, где Np это количество пикселей в строке. Регистр HDTIM задает начальный номер счетчика, после которого начинается выдача пиксельной информации, а также номер счетчика после которого прекращается выдача информации. Таким образом будет задана активная область горизонтальной строки.

Таблица 117 – Управление горизонтальной активной областью панели (HDTIM)

Бит	Имя	Функция
9:0	hdps	Начальная позиция активной области (hdps+1)
25:16	hdpe	Конечная позиция активной области (hdpe+1)

## 17.9 Управление вертикальной активной областью панели (VDTIM)

Этот регистр осуществляет управление видимой областью экрана и задает смещение данной области относительно начала экрана, а также ширину области. Единицей измерения является одна линия. Данный регистр задает номер линии, с которой начинается отображение информации на экране, а также номер линии, с которой начинается неотображаемая область. Ниже показаны функции, выполняемые регистрами при отображении информации (Рисунок 55).

Таблица 118 – Управление вертикальной активной областью панели (VDTIM)

Бит	Имя	Функция
9:0	vdps	Номер начальной отображаемой линии
25:16	vdpe	Номер конечной отображаемой линии

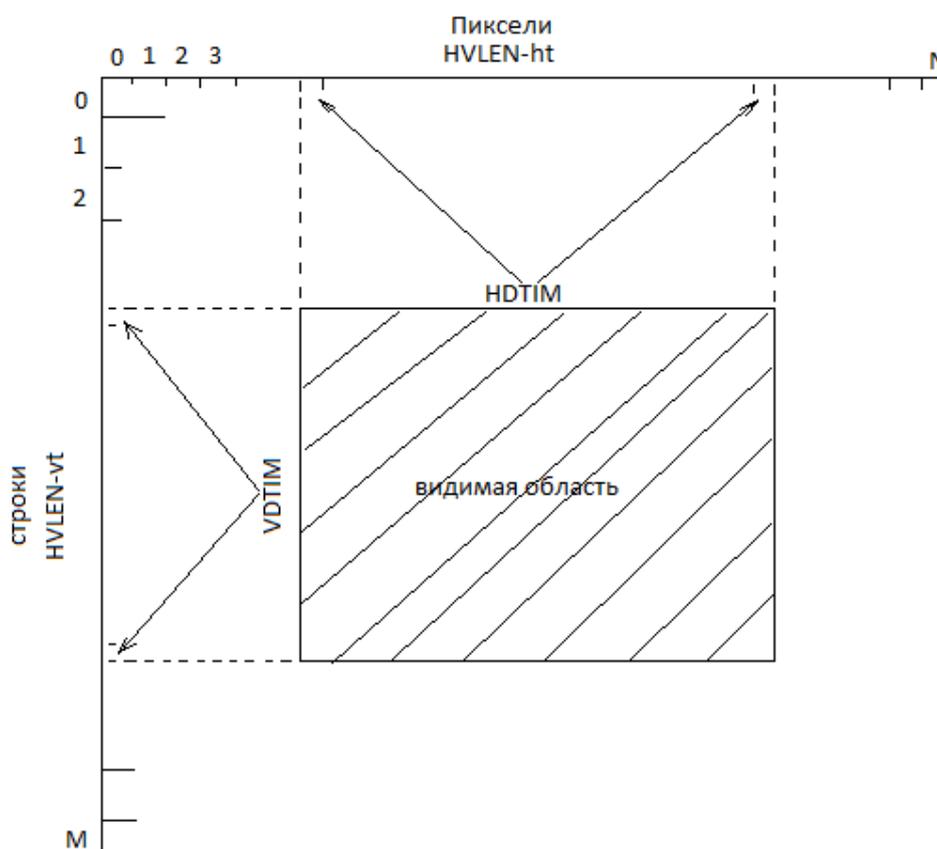


Рисунок 55 – Параметры отображаемой информации

### 17.10 Управление дополнительной горизонтальной активной областью панели (HDxTIM)

Этот регистр осуществляет управление дополнительной видимой областью линии и задает смещение данной области относительно начала строки, а также ширину области. Единицей измерения является длительность PX\_CLK сигнала. Регистр HDxTIM используется совместно с регистром HDTIM и позволяет задать пересечение двух активных областей. Информация будет выводиться только в ту область, которая принадлежит обоим множествам. В область, которая принадлежит множеству HDTIM, но не принадлежит множеству HDxTIM, будет выводиться значение фонового регистра FON.

Таблица 119 – Управление дополнительной горизонтальной активной областью панели (HDxTIM)

Бит	Имя	Функция
9:0	hxdps	Начальная позиция активной области (hxdps+1)
25:16	hxdpe	Конечная позиция активной области (hxdpe+1)

### 17.11 Управление дополнительной вертикальной активной областью панели (VDxTIM)

Этот регистр осуществляет управление видимой областью экрана и задает смещение данной области относительно начала экрана, а также ширину области. Единицей измерения является одна линия. Регистр VDxTIM используется совместно

с регистром VDTIM и позволяет задать пересечение двух активных областей. Информация будет выводиться только в ту область, которая принадлежит обоим множествам. В область, которая принадлежит множеству строк VDTIM, но не принадлежит множеству VDxTIM, будет выводиться значение фонового регистра FON.

**Таблица 120 – Управление дополнительной вертикальной активной областью панели (VDxTIM)**

Бит	Имя	Функция
9:0	vxdps	Номер начальной отображаемой линии
25:16	vxdpe	Номер конечной отображаемой линии

Регистры HDxTIM и VDxTIM могут быть использованы в случае, когда размер отображаемой информации значительно меньше разрешения панели. В этом случае данная функция позволяет значительно сократить трафик шины, т.к. передаваться будет только реально отображаемая информация.

## 17.12 Регистр FON

При использовании регистров задания дополнительного окна, данные выводятся только в область пересечения двух множеств: основного и дополнительного окон. В область основного окна, которая не входит в область дополнительного окна выводятся данные из регистра FON.

**Таблица 121 – Регистр FON**

Бит	Имя	Функция
7:0	B_fon	Данные для В выхода (используются биты 7:2)
15:8	G_fon	Данные для G выхода (используются биты 15:10)
23:16	R_fon	Данные для R выхода (используются биты 23:18)

## 17.13 Регистр конфигурации панели (PANEL\_CFG)

**Таблица 122 – Регистр конфигурации панели (PANEL\_CFG)**

Бит	Имя	Функция
5:0	-	
6	fpshi	Инверсия пиксель клона
8:7	-	
9	DVI_mode	При записи всегда должен быть 0.
10	DVI_clk	При записи всегда должен быть 0.
11	-	
31:12	-	Всегда ноль

## 17.14 Регистр управления выходом ШИМ (PWM\_CR)

Регистр может быть использован для управления яркостью панели.

**Таблица 123 – Регистр управления выходом ШИМ (PWM\_CR)**

Бит	Имя	Функция
0	clken	Разрешение работы (1)
2:1	-	Должен быть 0
3	frsh	Когда равен 1, на выходе PWM постоянный высокий уровень вне

		зависимости от работы внутренних счетчиков
7:4	Pwm_dv	делитель 0 – 1, 1 – 2, 2 – 4, 3 – 8, ....., 0xF - 32768
15:8	duty	Длительность активного уровня. 0 – всегда 0 1 – высокий для 1 периода 2 – высокий для 2 периодов ..... 0xFF - – высокий для 255 периодов
23:16	RELOAD	8-бит верхнее значение. Когда счетчик достигнет это значение, он загрузит ноль и начнет отсчет заново.

В качестве базовой частоты PWM модуля используется SOC\_CLK синхросигнал. Предварительно SOC\_CLK может быть поделен на число, задаваемое битами PWM\_dv. После чего мы будем иметь рабочую частоту PWM счетчика. Счетчик осуществляет увеличение на 1 каждый такт своей рабочей частоты. Если значение счетчика меньше значения “duty”, на внешнем выходе присутствует высокий уровень. В противном случае – низкий. При достижении счетчиком значения RELOAD, он сбросит свое значение в ноль и начнет счет заново.

### 17.15 Регистр управления сигналом GPIO\_0, 1, 2, 3

Регистр позволяет задать требуемую временную диаграмму на выводе GPIO\_0.

**Таблица 124 – Регистр управления сигналом GPIO\_0, 1, 2, 3**

Бит	Имя	Функция
9:0	Gp0_st	GPIO_0 начальная позиция (+1)
15	hpl	GPIO_0 активный уровень (0 – высокий, 1 – низкий)
25:16	Gp0_sp	GPIO_0 конечная позиция (+1)

Если значение начала импульса совпадает с его окончанием, сигнал изменяет свое значение на противоположное. Единицей измерений является пиксельклок.

Регистры GPIO\_1, 2, 3 аналогичны по функциям регистру GPIO\_0. Регистры позволяют сформировать, при необходимости, дополнительные управляющие сигналы.

### 17.16 Организация «спящего режима»

В обычном режиме работы, после включения контроллера, выполняется процесс генерации импульсов управления панелью, а также выдача информации. После выдачи одного фрейма данных (одной картины), контроллер обычно приступает к повторной выдаче новой или старой (если видеобuffer не обновлялся) информации. При этом ряд панелей требуют непрерывной подачи сигнала синхронизации и регенерации данных. Однако существуют панели, которые после выдачи одного фрейма данных, допускают останов синхронизации и некоторый перерыв в подаче информации. В этом случае контроллер после выдачи данных может перейти в спящий режим на некоторое время. Интервал, в течение которого контроллер не выполняет передачи данных, задается с помощью регистра SLP\_period. По истечении заданного периода контроллер возобновит чтение информации из видеобufferа и выдачу данных на внешний интерфейс. Переход в

режим сна выполняется при завершении формирования всех управляющих сигналов одного фрейма и при установленном бите `Slp_mode`. В режиме сна контроллер:

- Загружает в счетчик интервала сна значение регистра `SLP_period` и начинает обратный отсчет.
- Если бит `Slp_pclk` установлен в 1, выполняет останов делителя пиксельклока и таким образом останавливает синхронизацию всех внутренних счетчиков контроллера.
- Если бит `Slp_pxen` установлен в 1, запрещает работу всех внутренних счетчиков, даже если делитель пиксельклока работает (т.е. бит `Slp_pclk` равен 0). В этом случае активным может быть только выход `fpshift` (если бит `Slp_pclk` равен 0). Все другие выходы управления остановлены.
- Если бит `Slp_hold` установлен в 1, запрещает изменение счетчика пикселей и устанавливает его в нулевое значение. Аналогичен действию бита `hldv`.
- Если бит `Slp_clrf` установлен в 1, при завершении интервала сна, контроллер выполнит очистку всех флагов, которые были установлены после выгрузки предыдущего фрейма.

Когда счетчик интервала сна уменьшает свое значение до нуля, происходит выход из режима сна и контроллер возобновляет генерацию управляющих сигналов и выдачу информации. При реализации режима сна и установленном бите `Slp_clrf`, обязательно должен использоваться бит 12 (`hldm`). Бит `hldm` должен быть установлен в 1. Если бит `Slp_clrf` установлен в 1, при выходе из режима сна также инициализируется счетчик количества слов буфера видеоданных. Однако если бит `hldm` не будет установлен, то это может вызвать установку некорректного значения, т.к. к моменту окончания сна некоторое количество слов данных может быть прочитано (если канала DMA включен), обработано и размещено в выходном буфере.

Возможность использования спящего режима при работе контроллер, существенно снижает трафик к памяти процессора и сокращает потребление системы.

## 18 Интерфейс к аудио-кодеку AC97/I2S

Специальный интерфейс позволяет осуществить вывод аудио данных на внешний аудио-кодек типа AC97. Также данный интерфейс может работать по протоколу I2S шины и ее различных вариантов. Интерфейс имеет встроенные FIFO и может работать с DMA контроллером.

### 18.1 Регистры интерфейса AC97/I2S

Регистры интерфейса AC97/I2S подключены к пользовательской периферийной шине и имеют базовый адрес 0x8000\_0200(интерфейс 0) и 0x8000\_0220(интерфейс 1).

Таблица 125 – Регистры интерфейса

Номер	Название	Описание
0	SICR0	Регистр управления 0
1	SINT	
2	SICR2	Регистр управления 2
3	SISR	Регистр состояния после маскирования
4	SIRSR	Регистр состояния до маскирования
5	SIIER	Регистр разрешения прерывания
6	SIIDR	Регистр запрещения прерывания
7	SIICR	Регистр сброса флагов запросов прерывания
8	SIADR	Регистр-буфер аудио данных
9	SIMDR	Регистр-буфер модемных данных
10	ACCAR	AC канал, регистр адреса команды
11	ACCDR	AC канал, регистр данных команды
12	ACRAR	AC канал, регистр адреса состояния
13	ACSDR	AC канал, регистр данных состояния
14	ACGDR	AC канал, регистр данных GPIO
15	ACGSR	AC канал, регистр состояния GPIO
16	I2S_T_CR	Регистр управления передатчиком I2S
17	I2S_R_CR	Регистр управления приемником I2S
...	-	
28	SICR3	Регистр управления 3

#### 18.1.1 Регистр управления SICR0

Назначение разрядов регистра приведено ниже (Таблица 126).

Таблица 126 – Биты регистра управления SICR0

Бит	Имя	Функция
0	ENB	Включен (1) или выключен (0)
1	-	должен быть равен 0
2	MODEN	Модемный кодек присутствует (1)
3	AMCCH	АМС кодек присутствует (1)
4	RST	Запись 1 вызывает сброс внешнего кодека
5	LPBK	Тестовый режим (1)
6	BCKD	Источник синхронизации.

		Значение бита всегда должно быть равно 0, что означает внешний источник синхронизации. Внутренний источник не поддерживается.
7	MONO_DA	0 – стерео выход, 1 – моно выход
8	BIT8_DA	0 – 16 бит выход, 1 – 8 бит выход
9	MONO_AD	0 – стерео вход, 1 – моно вход
10	BIT8_AD	0 – 16 бит вход, 1 – 8 бит вход
31-11	-	

Возможно подключение только аудио кодека, отдельных аудио и модемного кодеков, совмещенного АМС кодека.

В тестовом режиме выход данных соединен с входом данных.

### 18.1.2 Регистр управления SICR2

Назначение разрядов регистра приведено ниже (Таблица 127).

**Таблица 127 – Биты регистра управления SICR2**

Бит	Имя	Функция
0	EREC	Разрешение записи (1)
1	ERPL	Разрешение воспроизведения (1)
2	EINC	Разрешение приема модемных данных (1)
3	EOUT	Разрешение передачи модемных данных (1)
4	EGPIO	Разрешение приема GPIO данных (1)
5	WKUP	Запись 1 вызывает инициализацию процедуры пробуждения кодека.
6	DRSTO	если 1 – выключает AC-link status read time-out function
22:7	REQLP	Коэффициент деления для формирования синхросигнала в режиме I2S
31:23	-	

### 18.1.3 Регистр управления SICR3

Назначение разрядов регистра приведено ниже (Таблица 128).

**Таблица 128 – Биты регистра управления SICR3.**

Бит	Имя	Функция
0	COMSEL	Разрешение вторичного кодека (1)
2:1	C2ID	ID вторичного кодека
3	DMA_EN	Разрешение работы встроенного контролера прямого доступа(1)
4	RELD	Разрешение перезагрузки после окончания текущего обмена (1)
5	TX/RX	Передача аудио данных(0), прием аудио данных(1)
31:6	-	

### 18.1.4 Регистр состояния SISR

Назначение разрядов регистра приведено ниже (Таблица 129).

**Таблица 129 – Биты регистра состояния SISR**

Бит	Имя	Функция
0	DTD	Data transfer done Устанавливается в 1 после завершения передачи в кодек адреса и данных. Сбрасывается в 0 посредством записи значения 1 в 0-й бит регистра SICR.

1	RDD	Read data done Устанавливается в 1 после завершения чтения из кодека адреса и данных. Сбрасывается в 0 посредством записи значения 1 в 1-й бит регистра SIICR.
2	GTD	GPIO data transfer done Устанавливается в 1 после завершения передачи в кодек GPIO данных. Сбрасывается в 0 посредством записи значения 1 в 2-й бит регистра SIICR.
3	-	
4	BSY	Busy 0 – интерфейс в нерабочем состоянии или выключен 1 – интерфейс передает или принимает фрейм в текущий момент
5	ATNE	Audio transmit FIFO not empty 0 – FIFO передатчика пусто 1 – FIFO передатчика содержит данные
6	ATNF	Audio transmit FIFO not full 0 – FIFO передатчика аудио данных заполнено полностью 1 – FIFO передатчика аудио данных заполнено не полностью
7	ARNE	Audio receive FIFO not empty 0 – в FIFO приемника нет данных 1 – в FIFO приемника есть данные
8	ATFS	Audio transmit FIFO service request 0 – FIFO передатчика аудио данных заполнено больше чем наполовину 1 – FIFO передатчика аудио данных заполнено меньше чем наполовину
9	ARFS	Audio receive FIFO service request 0 – FIFO приемника аудио данных заполнено меньше чем наполовину 1 – FIFO приемника аудио данных заполнено больше чем наполовину
10	ATUR	Audio transmit FIFO underrun Устанавливается в 1 при попытке чтения аудио данных из FIFO передатчика в момент, когда оно пусто. Сбрасывается в 0 посредством записи значения 1 в 10-й бит регистра SIICR.
11	AROR	Audio receive FIFO overrun Устанавливается в 1 при попытке записи аудио данных в FIFO приемника в момент, когда оно заполнено полностью. Сбрасывается в 0 посредством записи значения 1 в 11-й бит регистра SIICR.
12	MTNF	modem transmit FIFO not full 0 – FIFO передатчика модемных данных заполнено полностью 1 – FIFO передатчика модемных данных заполнено не полностью
13	MRNE	modem receive FIFO not empty 0 – в FIFO приемника нет данных 1 – в FIFO приемника есть данные
14	MTFS	modem transmit FIFO service request 0 – FIFO передатчика модемных данных заполнено больше чем наполовину 1 – FIFO передатчика модемных данных заполнено меньше чем наполовину
15	MRFS	modem receive FIFO service request 0 – FIFO приемника модемных данных заполнено меньше чем наполовину 1 – FIFO приемника модемных данных заполнено больше чем наполовину
16	MTUR	modem transmit FIFO underrun Устанавливается в 1 при попытке чтения модемных данных из FIFO передатчика в момент когда оно пусто. Сбрасывается в 0 посредством записи значения 1 в 16-й бит регистра SIICR.

17	MROR	modem receive FIFO overrun Устанавливается в 1 при попытке записи модемных данных в FIFO приемника в момент когда оно заполнено полностью. Сбрасывается в 0 посредством записи значения 1 в 17-й бит регистра SIICR.
18	RSTO	Read status time-out Устанавливается в 1 при чтении из кодека состояния time-out. Сбрасывается в 0 посредством записи значения 1 в 18-й бит регистра SIICR.
19	CLPM	Command for low-power mode равен 1 если частота кодека меньше частоты APB шины больше чем 25 раз. Детектирует отсутствие частоты синхронизации кодека.
20	CRDYPR	Primary codec ready
21	CRDYSC	Secondary codec ready
22	RESU	Возобновление Устанавливается в 1 при обнаружении sdata_in & EN_AC & CLPM. Сбрасывается в 0 посредством записи значения 1 в 22-й бит регистра SIICR
23	GINT	GPIO status change interrupt Устанавливается в 1 при приеме GPIO[0] = 1. Сбрасывается в 0 посредством записи значения 1 в 23-й бит регистра SIICR
24	RS3V	Receive slot 3 data valid Отражает значение бита достоверности 3-го слота.
25	RS4V	Receive slot 4 data valid
26	RS5V	Receive slot 5 data valid
27	RS12V	Receive slot 12 data valid
28	DMA_INT	Запрос на прерывание со стороны встроенного контроллера прямого доступа
31-29	-	

### 18.1.5 Регистр разрешения прерывания SIIER

Биты регистра разрешают запрос прерывания от соответствующих им разрядов регистра состояния. Запись 1 в соответствующий бит вызывает разрешение прерывания. Запись 0 не вызывает изменения разряда. Назначение разрядов регистра приведено ниже (Таблица 130).

**Таблица 130 – Биты регистра разрешения прерывания SIIER**

Бит	Имя	Разрешение прерывание если
0	DTD	Data transfer done
1	RDD	Read data done
2	GTD	GPIO data transfer done
7:3	-	
8	ATFS	Audio transmit FIFO service request
9	ARFS	Audio receive FIFO service request
10	ATUR	Audio transmit FIFO underrun
11	AROR	Audio receive FIFO overrun
13:12	-	
14	MTFS	modem transmit FIFO service request
15	MRFS	modem receive FIFO service request
16	MTUR	modem transmit FIFO underrun
17	MROR	modem receive FIFO overrun
18	RSTO	Read status time-out

21:19	-	
22	RESU	Resume.
23	GINT	GPIO status change interrupt
27-24	-	
28	DMA_INT	Разрешение прерывания от контроллера прямого доступа
31-29	-	

### 18.1.6 Регистр запрещения прерывания SIIDR

Биты регистра запрещают запрос прерывания от соответствующих им разрядов регистра состояния. Запись 1 в соответствующий бит вызывает запрещение прерывания. Запись 0 не вызывает изменения разряда. Назначение разрядов регистра аналогично разрядам регистра SIIER.

### 18.1.7 Регистр адреса команды ACCAR

При программировании регистров внешнего аудио-кодека необходимо задать адрес регистра и записываемые данные. Адрес регистра кодека записывается в регистр ACCAR интерфейса.

Назначение разрядов регистра приведено ниже (Таблица 131).

**Таблица 131 – Биты регистра адреса команды ACCAR**

Бит	Имя	Функция
11:0	-	Не используются
18:12	IX	Индекс управляющего регистра кодека. Бит 12 всегда должен быть равен 1.
19	RW	Бит чтения (1) или записи (0)
31:20	-	Не используются

### 18.1.8 Регистр данных команды ACCDR

При программировании регистров внешнего аудио-кодека необходимо задать адрес регистра и записываемые данные. Данные регистра кодека записывается в регистр ACCDR интерфейса.

Назначение разрядов регистра приведено ниже (Таблица 132).

**Таблица 132 – Биты регистра данных команды ACCDR**

Бит	Имя	Функция
3:0	-	Не используются
19:4	CDR	16 бит записываемые данные.
31:20	-	Не используются

### 18.1.9 Регистр состояния адреса команды ACSAR

Назначение разрядов регистра приведено ниже (Таблица 133).

**Таблица 133 – Биты регистра состояния адреса команды ACSAR**

Бит	Имя	Функция
11:0	-	Не используются
18:12	SAR	Эхо индекса управляющего регистра кодека, данные которого были прочитаны.
31:19	-	Не используются

### 18.1.10 Регистр состояния данных команды ACSDR

Назначение разрядов регистра приведено ниже (Таблица 134).

**Таблица 134 – Биты регистра состояния данных команды ACSDR**

Бит	Имя	Функция
3:0	-	Не используются
19:4	SDR	16 бит данных, прочитанных из кодека
31:20	-	Не используются

### 18.1.11 Регистр данных GPIO ACGDR

Имеется возможность управлять некоторыми линиями общего назначения кодека. Данные, для записи в биты GPIO кодека должны быть предварительно записаны в регистр ACGDR.

Назначение разрядов регистра приведено ниже (Таблица 135).

**Таблица 135 – Биты регистра данных GPIO ACGDR**

Бит	Имя	Функция
3:0	-	Не используются
19:4	GDATA	16 бит данных, которые будут записаны в GPIO кодека
31:20	-	Не используются

### 18.1.12 Регистр состояния данных GPIO ACGSR

Назначение разрядов регистра приведено ниже (Таблица 136).

**Таблица 136 – Биты регистра данных GPIO ACGSR**

Бит	Имя	Функция
3:0	-	Не используются
19:4	SDR	16 бит GPIO данных, прочитанных из кодека
31:20	-	Не используются

### 18.1.13 Регистр аудио данных SIADR

Этот регистр используется для записи аудио отсчетов, передаваемых во внешний аудио-кодек при воспроизведении звука. Этому регистру соответствует FIFO глубиной восемь 32-разрядных слов. Формат записываемых данных зависит от режима МОНО или СТЕРЕО, а также от 8 или 16 бит разрешения:

- для 16-бит стерео данных младшие 16 бит регистра данных соответствуют правому каналу, а старшие – 16 бит левому;
- для 16-разрядного моно режима младшая половина хранит текущий отсчет, а старшая – следующий за ним;
- для 8-разрядного стерео: биты 7:0 – правый канал N, биты 15:8 – правый канал N+1, биты 23:16 – левый канал N, биты 31:24 – левый канал N+1;
- для 8-разрядного моно: биты 7:0 – отсчет N, биты 15:8 – отсчет N+1, биты 23:16 – отсчет N+2, биты 31:24 – отсчет N+3.

При чтении регистра данных, данные считываются из FIFO приемника. FIFO приемника имеет глубину восемь 32-битных слов.

### **18.1.14 Регистр модемных данных SIMDR**

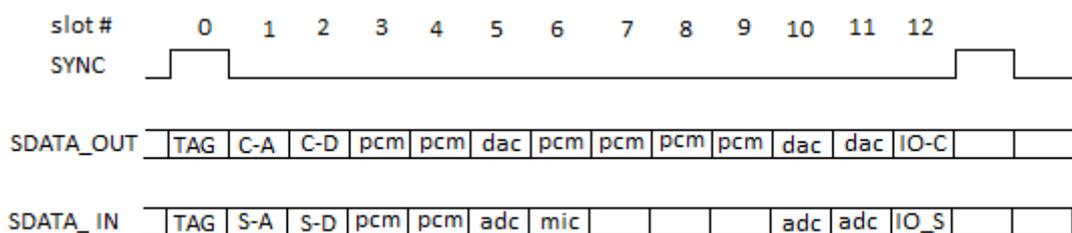
Этот регистр используется для записи данных, передаваемых во внешний модем. Этому регистру соответствует FIFO глубиной восемь 16-разрядных слов. При чтении данные берутся из FIFO приемника модема, имеющего глубину восемь 16-бит слов.

## **18.2 Режимы работы**

### **18.2.1 Режим работы AC97**

Интерфейс имеет возможность поддержки стандартов AC97(только аудио), MC97(только модем), AMC97(аудио и модем совмещены). AC97 и AMC97 могут работать только как первичные (primary), MC97 может быть сконфигурирован как первичный и как вторичный (secondary). Протокол AC-канала определяет полнодуплексный последовательный интерфейс между внешним кодеком и контроллером. Внешний кодек (или контроллер, если нет первичного кодека в системе) генерирует 12,288 МГц частоту, поступающую в контроллер. В контроллере входная частота делится на 256, получая, таким образом, 48 КГц звуковую частоту. Она используется контроллером и отсылается в кодек как сигнал SYNC.

Передача данных между кодеком и интерфейсом осуществляется в виде фреймов. Один фрейм содержит 13 слотов (Рисунок 56). Каждый слот (исключая нулевой) содержит 20 бит последовательных данных. Сигнал SYNC определяет начало фрейма. Он активен (высокий уровень) только в течение нулевого слота (начальный слот). Нулевой слот имеет длительность 16 бит. Хотя каждый слот имеет длительность 20 бит, только 16 бит используется.



**Рисунок 56 – Структура фрейма**

Таким образом, один фрейм содержит 256 бит информации (20•12+16). Сигнал SYNC формируется контроллером и имеет частоту следования равную 12,288/256 = 48 КГц. Длительность высокого уровня сигнала SYNC равна 16-ти периодам сигнала синхронизации. Структура тэга фрейма и последовательность бит слотов показана ниже (Рисунок 57).

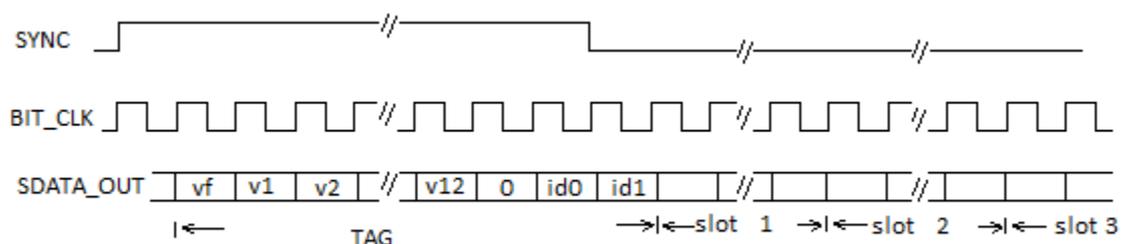


Рисунок 57 – Структура тега

Фрейм имеет бит достоверности – vf. Также имеют биты достоверности и все 12 последующих слотов. Контроллер имеет доступ к регистрам внешнего кодека посредством отсылки адреса регистра в слоте 1 и данных в слоте 2. Аудио данные передаются в 3, 4 и 5 слотах. Имеется возможность управлять GPIO выводами внешнего кодека, а также анализировать их состояние. Для этого используется последний 12 слот. Формат обмена полнодуплексный, т.е. одновременно с передачей по линии SDATA\_OUT контроллер принимает данные по линии SDATA\_IN. Для ознакомления с подробным описанием каждого слота необходимо изучение спецификации AC97.

Контроллер имеет возможность принимать сообщение о пробуждении (wake-up) внешнего кодека и генерировать соответствующий запрос прерывания.

## 18.2.2 Режим работы I2S

Интерфейс имеет возможность поддержки стандарта I2S. Для этих целей предусмотрены два управляющие регистра.

### 18.2.2.1 Регистр управления передатчиком I2S\_T\_CR

Таблица 137 – Биты регистра управления передатчиком I2S\_T\_CR

Бит	Имя	Функция
0	TEN	Разрешение работы (1)
1	MODE	0 – I2S режим 1 – DSP режим
2	SONY	Выбор стандарта I2S : SONY(1) или Philips (0) Имеет смысл только в режиме I2S
3	MS	Master (1) or slave(0) Бит определяет того, кто генерирует синхросигнал и строб начала фрейма (левый-правый)
9:4	DSS	Длина передаваемых данных – от 1 до 64 бит. Значение в битах DSS всегда на 1 меньше количества бит передаваемых данных.
10	PNIS	Определяет момент приема выбора канала: 0 – прием по положительному фронту (из 0 в 1); 1 – прием по отрицательному фронту
11	PNOS	Определяет момент выдачи данных и выбора канала: 0 – выдача по положительному фронту (из 0 в 1); 1 – выдача по отрицательному фронту.
12	SWHW	1 – указывает на необходимость перестановки 16-разрядных полуслов в одном слове

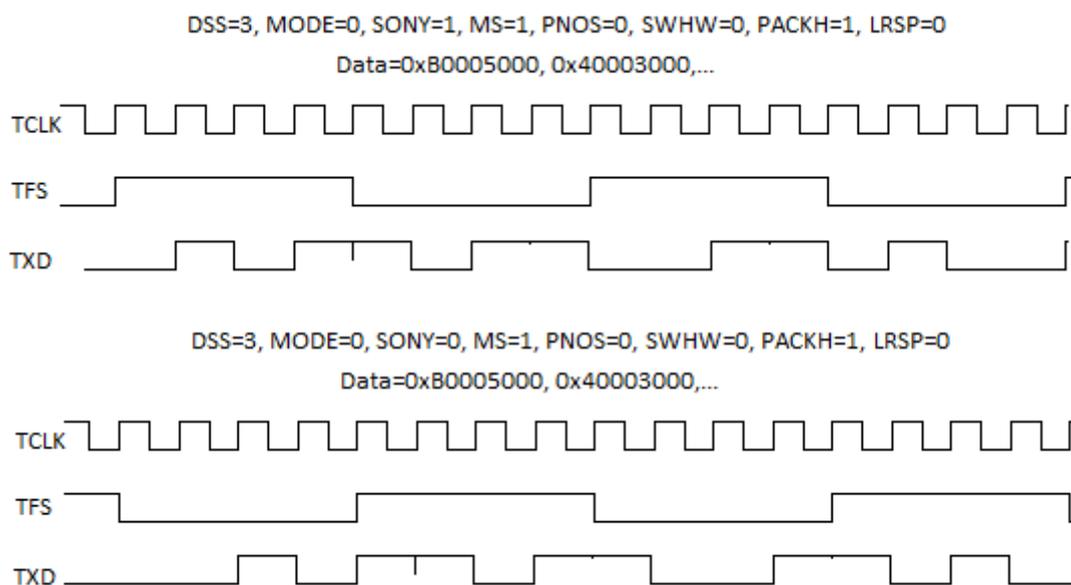
13	PACKH	Имеет смысл при длине передаваемых данных 16 бит и меньше. 1 – указывает на то, что в одном 32-разрядном слове упакованы два значения. Иначе в одном слове хранится одно значение.
14	LRSP	1 и установлен DSP-режим– указывает на то, что общее отсылаемое слово состоит из двух симметричных частей (левой и правой) которые размещаются в разных 32-бит словах. 0 – указывает на то, что передаваемая последовательность не делится на части и представляет собой непрерывный поток бит
15		

Регистр передатчика имеет длину 32 бита. Выдача бит происходит всегда «старший бит первый». При передаче данных, длина которых меньше 32 бит, данные необходимо выровнять влево. Если данные упакованы в полуслова, то при передаче данных длиной менее 16 бит их необходимо выровнять влево до границы полуслова.

В режиме I2S передатчик не может передавать данные длиной более 32 бит. При включенном передатчике передача данных инициируется импульсом начала фрейма (DSP режим) или изменением выбора канала (I2S режим). В случае если новые данные отсутствуют в буфере передатчика, происходит повторная передача последних переданных данных.

В режиме DSP можно передавать данные длиной более 32 бит. При этом эти данные могут состоять из равных половин (левой и правой), если LRSP == 1, или представлять собой непрерывный поток бит расположенных в последовательных словах.

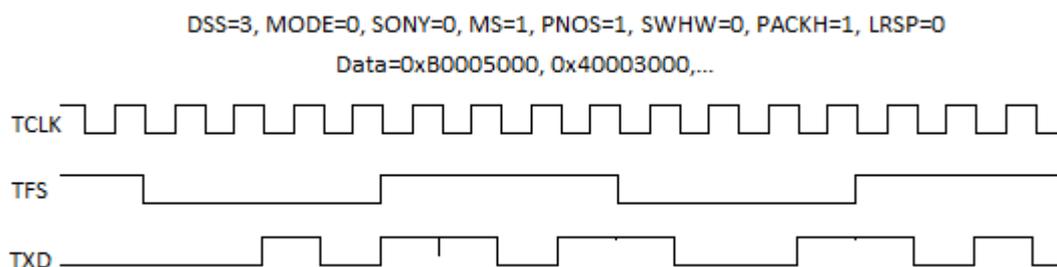
Ниже представлен пример передачи 4-разрядных данных в режиме мастера (Рисунок 58).



**Рисунок 58 – Передача 4-разрядных данных**

Передача происходит в режиме I2S (бит MODE=0). Линия TFS указывает на левый или правый канал передачи. Вся информация выдается относительно положительного фронта синхросигнала. Видим, что данные упакованы по 16 бит в одно 32-разрядное слово и выравнены по направлению к старшему бит. Основное отличие между режимами SONY и Philips состоит в том, что в режиме Philips данные

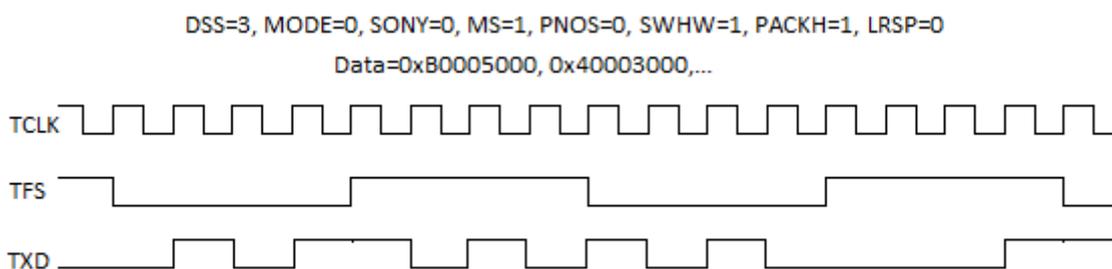
выдаются с задержкой на один такт относительно импульса начала выбора канала. Если установить бит PNOS в 1, выдача сигналов выбора канала и данных будет происходить относительно отрицательного фронта синхросигнала. Пример приведен ниже (Рисунок 59).



**Рисунок 59 – Выдача информации по отрицательному фронту TCLK**

Это позволяет более гибко адаптироваться при подключении к приемнику информации и обеспечить надежное время предустановки и удержания данных.

Установка бита SWHW в 1 вызовет перестановку полуслов в слове данных. Пример приведен ниже (Рисунок 60). Данный прием позволяет поменять местами левый и правый каналы.



**Рисунок 60 – Перестановка полуслов в слове**

Интерфейс может работать в модифицированном режиме или режиме DSP. Основное отличие данного режима состоит в формировании импульса TFS. В режиме DSP он не переключается между правым и левым каналом, а импульсом информирует о начале нового фрейма. Пример приведен ниже (Рисунок 61).

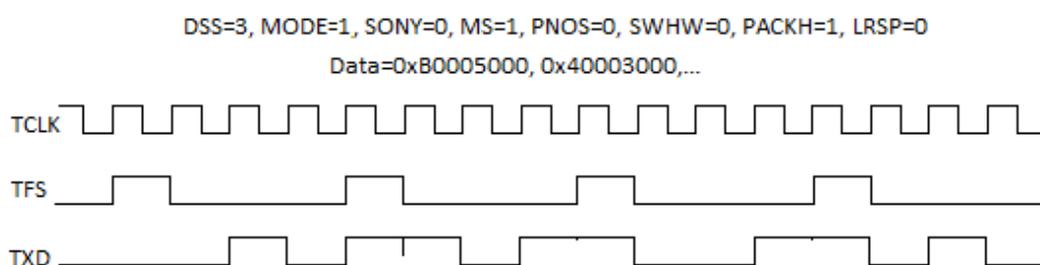


Рисунок 61 – Режим передачи DSP

Импульс на выходе TFS указывает, что в следующем такте начинается выдачи информации нового фрейма. Каждые 4 бита данных сопровождаются импульсом начала фрейма (Рисунок 61). В данном случае уже нет деления на левый и правый каналы. Однако если необходимо передать информацию, состоящую из двух частей, то возможны два способа. Первый состоит в упаковке двух половин информации в одно 32-разрядное слово и передаче как единого 32-разрядного фрейма. Второй способ может использоваться, если две порции информации находятся в последовательных словах, следующих друг за другом. В этом случае необходимо установить бит LRSP в 1. Для нашего примера передачи данных временная диаграмма представлена ниже (Рисунок 62).

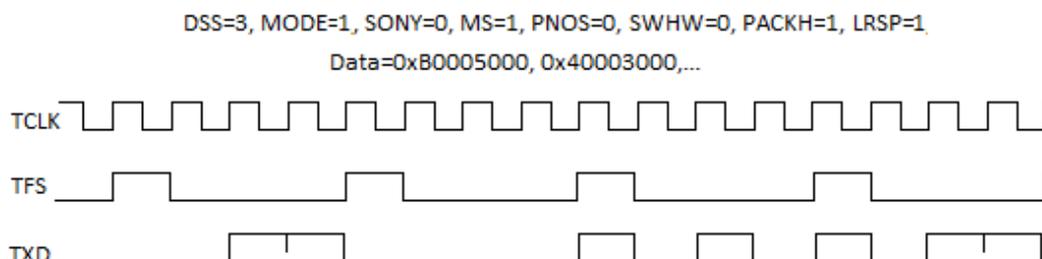


Рисунок 62 – Режим передачи смежными словами

В случае, когда информация находится в смежных словах (в нашем случае в смежных полусловах, т.к. длина данных меньше 16 бит и установлен признак PACK) количество передаваемых бит делится пополам и сначала передаются биты одной половины, а затем второй. В нашем случае передается 4 бита, поэтому сначала будут переданы два старшие бита младшего полуслова (биты 01b от значения 5), а затем два старшие бита старшего полуслова (биты 10b от значения В). Данный режим обычно используется, если длина объединенной посылки превышает 32 бита и в одном слове невозможно упаковать две передаваемые части.

Бит MS в регистре управления определяет источник формирования сигнала синхронизации и сигнала выбора канала (строб начала фрейма). Если бит установлен в 1, передатчик формирует данные сигналы. В противном случае внешнее устройство подает сигналы синхронизации и выбора канала. Последний случай обычно используется, если внешнее устройство требует формирования заданной частоты, которая не может быть получена в процессоре путем деления частоты SOC-шины на константу. Отметим, что в случае работы в режиме подчиненного (MS=0) интерфейс принимает сигнал выбора канала. В этом случае имеет значение бит PNIS, т.к. он задает момент приема достоверного значения сигнала выбора канала.

### 18.2.2.2 Регистр управления приемником I2S\_R\_CR

Таблица 138 – Биты регистра управления приемником I2S\_R\_CR

Бит	Имя	Функция
1	MODE	0 – I2S режим 1 – DSP режим
2	SONY	Выбор стандарта SONY (1) или Philips (0)

		Имеет смысл только в режиме I2S
3	MS	Master (1) or slave (0) Бит определяет того кто генерирует синхросигнал и строб начала фрейма (выбор канала левый-правый)
9:4	DSS	Длина передаваемых данных – от 1 до 64 бит
10	PNIS	Определяет момент приема данных и выбора канала: 0 – прием по положительному фронту (из 0 в 1); 1 – прием по отрицательному фронту.
11	PNOS	Определяет момент выдачи выбора канала: 0 – выдача по по положительному фронту (из 0 в 1); 1 – выдача по отрицательному фронту.
12	SWHW	1 – указывает на необходимость перестановки 16-разрядных полуслов в одном слове
13		
14	LRSP	1 и установлен DSP режим – указывает на то, что общее принимаемое слово состоит из двух симметричных частей (левой и правой) которые необходимо принять отдельно в разные 32-разрядные слова. 0 – указывает на то, что принимаемая последовательность не делится на части и представляет собой непрерывный поток бит

В режиме I2S используются FIFO передатчика и приемника аудио данных. Поддержка модемного кодека невозможна.

В случае приема данных в режиме I2S, если длина данных меньше или равна 16 бит, данные будут упаковываться в 32-разрядные слова. При этом возможна перестановка полуслов в слове. В случае длины принимаемых данных больше 16 бит, данные помещаются в 32-разрядные слова. При этом достоверны только DSS+1 младших бит 32 разрядного слова. В случае приема данных, длина которых больше 32 бит, возможны два варианта приема:

- 1 если LRSP == 0, первые 32 бита будут размещены в первом слове, а оставшиеся биты в следующем слове;
- 2 если LRSP == 1, принимаемые данные делятся поровну на левую и правую половины, каждая из которых размещается в индивидуальном слове.

Прием данных и выбора слова может происходить как по отрицательному фронту синхросигнала, так и по положительному. Выбор фронта программируется.

Как приемник, так и передатчик в режиме мастер могут выдавать наружу сигнал синхронизации обмена и строб начала обмена. Для задания частоты обмена используется внутренний делитель. Базовой частотой делителя является клок периферийной шины. Значение коэффициента деления находится в регистре SICR2[22:7]. Значение коэффициента деления на 1 больше записываемого значения.

Биты разрешения работы только включают передатчик или приемник. Для того чтобы полностью включить прием или/и передачу необходимо установить соответствующие биты в регистре SICR2[1:0]. Если приемник или передатчик выключен, то все его внутренние регистры сбрасываются в ноль. Очень важным для корректного начала работы может быть необходимость установки FIFO передатчика и приемника в начальное состояние. Это можно сделать битом SICR0[4], который производит сброс всех FIFO интерфейса (указателей и флагов) в исходное состояние. Для сброса бит нужно установить, а затем программно очистить. После этого FIFO передатчика можно заполнить значением 0, если передаваемые данные еще не готовы.

## 19 ARINC контроллер

Контроллер интерфейса ARINC содержит в своем составе восемь приемников и четыре передатчика по ГОСТ 18977-79 (далее ARINC).

Каждый приемник поддерживает функцию распознавания меток (или адресов). Для каждого приемника может быть запрограммировано до 256 меток. Помимо этого фильтрация входных данных может осуществляться не только на базе меток, но и на базе двух бит Источник/Приемник.

Каждый передатчик поддерживает однонаправленную передачу 32-х разрядных слов по двухпроводной витой паре, используя формат кодирования RZ. Доступна возможность запрограммировать 32-й бит как данные или как бит паритета. В случае формирования бита паритета, программируется его четность или нечетность.

Каждый приемник и передатчик использует собственный буфер FIFO для хранения данных. Размеры буфера FIFO варьируются от 32x32 до 256x32. Статус наполненности FIFO определяется на основе соответствующих бит статуса для каждого FIFO. Контроллер поддерживает различные скорости приема и передачи данных. Работа контроллера осуществляется на базовой частоте 1 МГц, что позволяет обнаруживать ошибки в скорости приема/передачи данных, а также в паузах между сообщениями.

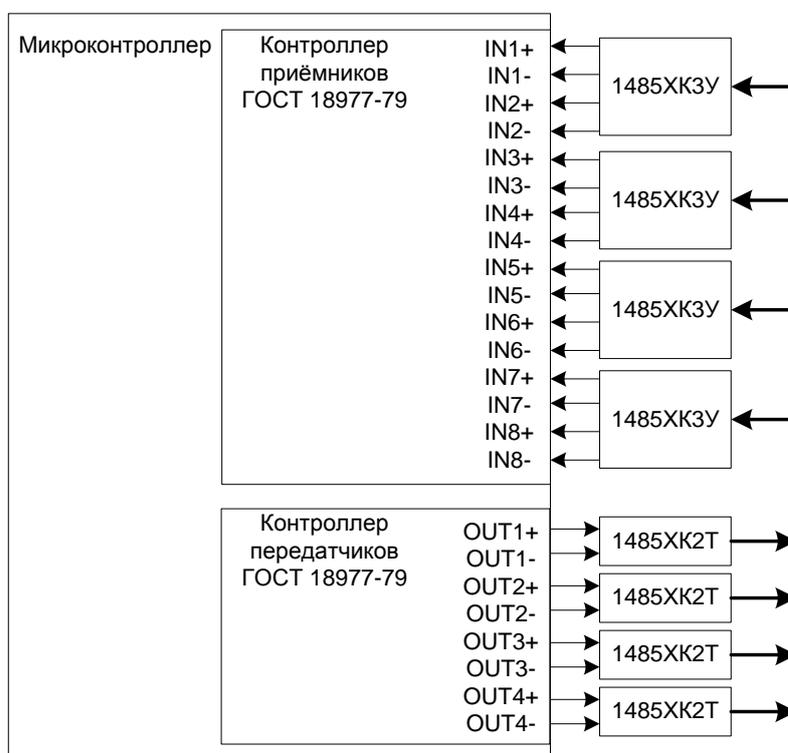


Рисунок 63

Особенности:

- симплексный режим приема/передачи со скоростями 12,5 кГц или 100 кГц;
- фильтрация входных данных на базе 256 меток и двух бит Источник/Приемник для каждого приемника;
- возможность передачи 32 бита, как данных, так и паритета;
- выбор четности/нечетности бита паритета;

- размеры буферов FIFO передатчиков: одно 256x32, три 64x32;
- размеры буферов FIFO приемников: два 256x32, четыре 64x32, два 32x32;
- возможность формирования прерываний при разных статусах наполненности буферов FIFO и при возникновении ошибок скорости передачи слова и паузы между словами;
- маскирование прерываний.

## 19.1 Формат слова

Слова в интерфейсе ARINC всегда 32-разрядные, и включают в себя пять полей: паритет, SSM, данные, источник/приемник, метка. Биты передаются младшими разрядами вперед, за исключением метки, которая передается старшими разрядами вперед. В результате можно описать порядок следования бит по шине ARINC следующим образом:

8, 7, 6, 5, 4, 3, 2, 1, 9, 10, 11, 12, 13...32.

32	31	30	29	11	10	9	8	1
P	SSM		DATA MSB	SDI		LABEL		
			LSB					

Старший разряд всегда бит паритета. Стандартом установлено, что бит паритета должен дополнять слово до нечетного. Таким образом, количество единиц в 32-разрядном слове должно быть нечетным. Например, если биты 1-31 содержат четное количество единиц, бит паритета должен быть установлен в единицу, с другой стороны, если биты 1-31 содержат нечетное количество единиц, то бит паритета должен быть сброшен в ноль.

Биты 31 и 30 содержат знак или статус. В контроллере эти биты рассматриваются как обычные данные и помещаются в FIFO вместе с полем данных без изменений и дополнительной обработки.

Как пример биты 31 и 30 могут кодировать следующие характеристики, представленные ниже (Таблица 139).

**Таблица 139 – Биты 31 и 30**

Бит		Значение
31	30	
0	0	плюс, север, восток, справа, к, выше
0	1	невывислительные данные
1	0	функциональный тест
1	1	минус, юг, запад, слева, от, ниже

Биты 10 и 9 позволяют распознать Источник/Приемник данных. Это применяется при нескольких приемниках на шине ARINC, чтобы определить, для кого из них предназначаются данные. В системе со сложной структурой эти биты могут также использоваться, чтобы определить источник передачи. В остальных случаях эти разряды используются как данные. Следует отметить, что в интерфейсе ARINC на одной витой паре может быть один передатчик и до 20 приемников. Если включена функция проверки этих бит, при их несовпадении с битами, заданными программно в контроллере, сообщение не будет помещено в FIFO.

Биты с 1 по 8 позволяют идентифицировать тип данных оставшейся части слова, следовательно, методы преобразования, применяемые к данным. Помимо этого в контроллере метки используются для фильтрации входных данных, то есть если метка в принятом сообщении не соответствует ни одной из меток

определенной в памяти меток приемного канала, то данные не помещаются в FIFO. Это может служить аналогом того, что приемник не может интерпретировать метод обработки этих данных, следовательно, эти данные предназначены для другого приемника.

В случае если приемник принимает данные с неправильным битом паритета, они также не будут помещены в FIFO.

## 19.2 Структурная схема канала приема

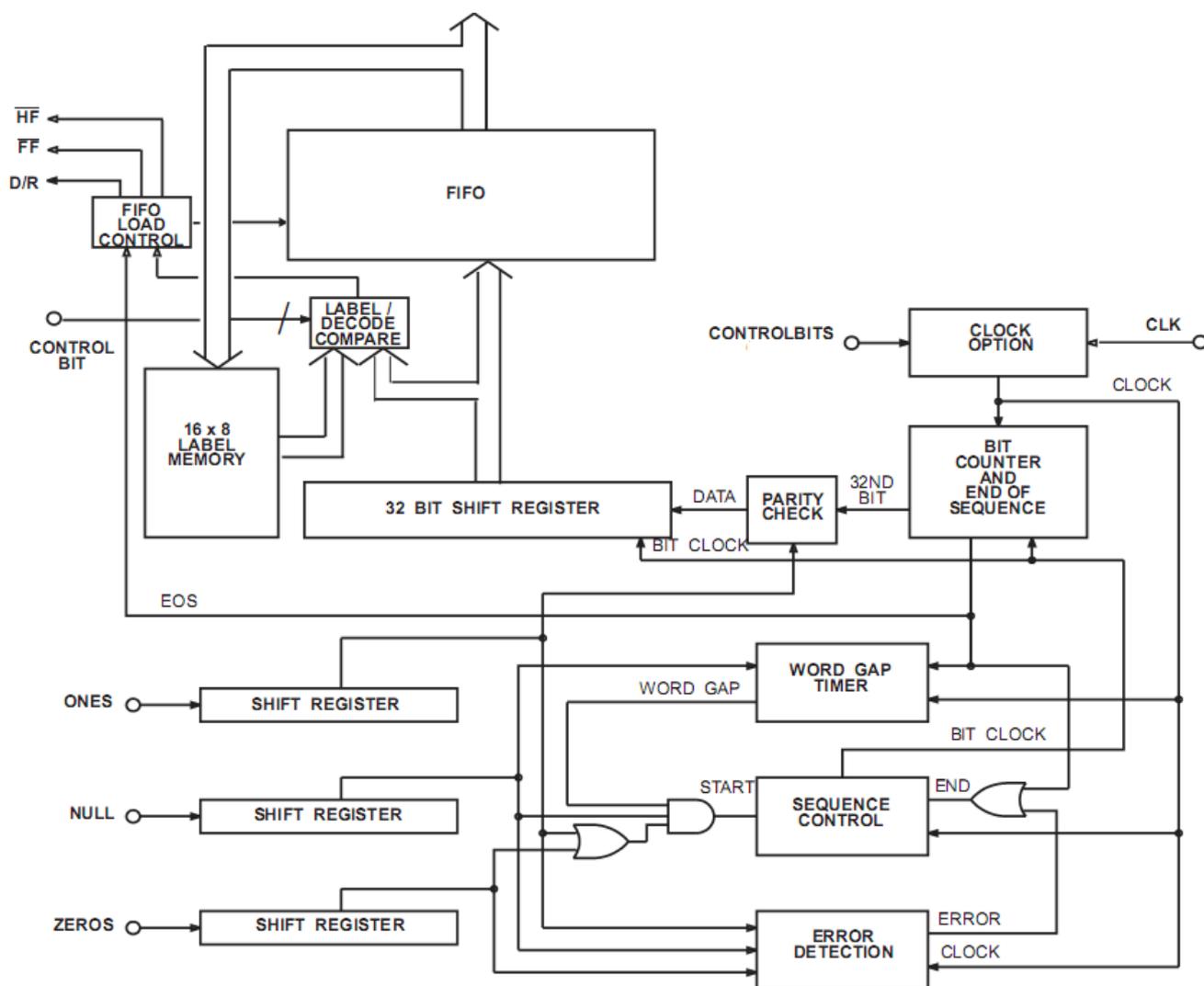


Рисунок 64 – Структурная схема канала приема

Представленная на рисунке схема работает на частоте CLK = 1 МГц, в этом случае ошибка обнаружения бита в линии не будет составлять более 0,1 %.

Сдвиговые регистры длиной 10 бит, предназначенные для обнаружения в линии трех последовательностей единиц (Ones), нулей (Zeros) и отсутствие сигнала (Null), позволяют считать данные действительными. В дополнении к этому для бит данных, One или Zero в верхних битах сдвигового регистра должны сопровождаться Null в нижних битах в пределах битового интервала. В пределах паузы между сообщениями, три последовательных бита Null должны быть сэмплированы в верхней и нижней части сдвигового регистра Null. В этом случае гарантируется минимальная ширина импульса данных.

Каждый бит данных должен быть обнаружен в пределах от 8 до 12 сэмплов. В этом случае скорость передачи считается верной.

Таймер паузы между сообщениями сэмплирует сдвиговый регистр Null каждые 10 входных тактов (или 80 тактов для скорости 12,5 кГц) после последнего полученного бита данных. Если Null обнаружен, таймер инкрементируется. Значение таймера равное трем разрешает следующий прием.

Схема паритета считает количество принятых единиц, включая бит паритета. Если результат нечетный, на выходе схемы формируется сигнал равный нулю.

После того как приняты все 32 бита логика приемника формирует сигнал конец последовательности (EOS). В зависимости от состояния бит LB\_EN, SD\_EN, SDI1, SDI2 регистра управления принимается решение о загрузке принятых данных в FIFO. Если в принятом слове биты 9 и 10 не соответствуют правилам или не совпала метка, слово не загружается в FIFO. Случаи, в которых происходит загрузка FIFO принятыми данными, перечислены ниже (Таблица 140).

**Таблица 140 – Загрузка FIFO принятыми данными**

LB_EN	Результат сравнения слова ARINC с меткой	SD_EN	Результат сравнения бит 9,10 слова ARINC с SDI1, SDI2	FIFO
0	X	0	X	Загружается
1	не совпала	0	X	Игнорируются
1	совпала	0	X	Загружается
0	X	1	не совпали	Игнорируются
0	X	1	совпали	Загружается
1	совпала	1	не совпали	Игнорируются
1	не совпала	1	совпали	Игнорируются
1	не совпала	1	не совпали	Игнорируются
1	совпала	1	совпали	Загружается

Если хотя бы одно слово загружено в FIFO, устанавливается в единицу сигнал DR, что отражается в регистре статуса контроллера. Флаг остается в неизменном состоянии, пока последнее слово не будет прочитано из FIFO и оно не будет пустым. Помимо этого применяются еще два сигнала характеризующие состояние FIFO, а именно HF означает, что FIFO наполовину полно и FF означает, что FIFO полно. Установка этих сигналов также отражается в регистре статуса. Каждый из этих флагов может быть источником прерывания, в случае если оно разрешено соответствующим битом маскирования регистра управления.

### **19.3 Структурная схема канала передачи**

Если флаг TX\_R в состоянии логической единицы, это значит что FIFO пусто и в него могут быть загружены 31- или 32-битные данные. Количество слов данных определяется размером FIFO для выбранного канала передачи. Если флаг TX\_R в состоянии логического нуля, тогда только в доступные в FIFO ячейки можно загрузить данные. Если FIFO заполнено полностью, флаг FFT установлен в единицу, то FIFO игнорирует дальнейшие попытки записи в него. FIFO наполовину пусто, если установлен флаг HFT, в этом случае можно загрузить данными оставшуюся половину буфера FIFO.

В нормальном режиме работы 32 бит передаваемых данных является битом паритета. Четность или нечетность выбирается битом ODD регистра управления. Если бит разрешения паритета (EN\_PAR) сброшен в ноль, тогда 32 бит передается, как бит данных из FIFO.

Если бит CH\_EN установлен в единицу и FIFO передачи не пусто, начинается передача слов данных из FIFO до тех пор, пока FIFO не будет пусто или не будет сброшен бит CH\_EN.

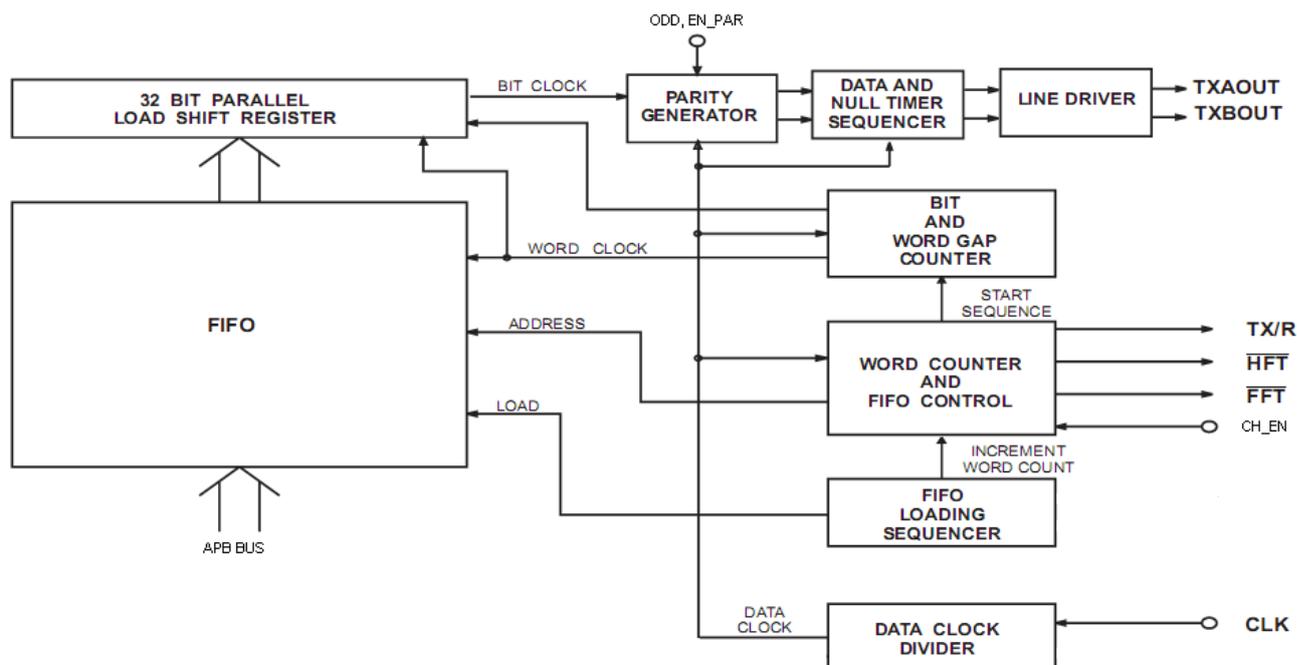


Рисунок 65 – Структурная схема канала передачи

## 19.4 Описание регистров приемника

Таблица 141 – Регистры приемника

Базовый адрес	Номер	Название	Состояние после сброса	Описание
0x8000_2000		ARINC429R		
+0				Канал 0
	0	RX_CR		Регистр управления приемника
	1	-		-
	2	DATA_R		FIFO принимаемых данных
	3	STATUS		Регистр состояния приемника
+4				Канал 1
+8				Канал 2
+12				Канал 3
+16				Канал 4
+20				Канал 5
+24				Канал 6
+28				Канал 7
0x8000_2400		Память приемников		Прямой доступ в память приемников
	0x0000-0x00FF	FIFO_D0		Буфер приемника канала 0
	0x0100-0x01FF	FIFO_D1		Буфер приемника канала 1

	0x0200-0x023F	FIFO_D2		Буфер приемника канала 2
	0x0240-0x027F	FIFO_D3		Буфер приемника канала 3
	0x0280-0x02BF	FIFO_D4		Буфер приемника канала 4
	0x02C0-0x02FF	FIFO_D5		Буфер приемника канала 5
	0x0300-0x031F	FIFO_D6		Буфер приемника канала 6
	0x0320-0x033F	FIFO_D7		Буфер приемника канала 7
	0x0340-0x03BF	-		-
	0x03C0-0x03FF	LABELS		Буфер меток. Для каждого канала от 0 до 7 используется 8 32-разрядных слов образующих одно 256 битовое кодовое слово метки. Каждый бит соответствует одному из 256 возможных значений метки. Номер слова – три младших бита адреса. Номер канала – биты 5..3 адреса.

Доступ к буферу данных возможен как прямым, так и косвенным путем (через указатель чтения). Доступ к меткам только прямым путем.

## 19.5 Регистр управления приемника RX\_CR

**Таблица 142 – Биты регистра управления приемника RX\_CR**

Бит	Имя	Значение	Описание
0	CH_EN		Разрешение работы канала 1 – прием по каналу разрешен 0 – канал приема находится в состоянии сброса
1	CLK		Скорость приема данных 1 – частота приема данных = опорная частота/80 (12,5 кГц) 0 – частота приема данных = опорная частота/10 (100 кГц)
2	LB_EN		Разрешение обнаружения меток 1 – разрешено обнаружение меток в первых 8 принятых битах 0 – обнаружение отключено, все принятые данные помещаются в FIFO
3	SD_EN		Разрешение декодирования бит данных 9 и 10 1 – разрешено сравнение бит данных 9 и 10 со значением бит SDI1 и SDI2 соответствующего канала 0 – декодирование отключено, все принятые данные помещаются в FIFO
4	DA		Бит прямого доступа 1 – память приема канала работает не в режиме FIFO, доступ к ней осуществляется в диапазоне адресов в зависимости от номера канала 0 – обычный режим работы FIFO При приеме данных из канала занесение их в память происходит в соответствии с адресом в первых восьми битах сообщения.

5	SDI1		Бит сравнения SDI1 Значение бита сравнивается с битом 9 принимаемых данных, если установлен бит SD_EN соответствующего канала
6	SDI2		Бит сравнения SDI2 Значение бита сравнивается с битом 10 принимаемых данных, если установлен бит SD_EN соответствующего канала
7	ENSYNC		Разрешение работы входов приемника в режиме данных и синхросигнала 1 - разрешено 0 – запрещено При установленном бите ENSYNC для соответствующего канала вход IN_A работает как данные (D), вход IN_B как синхросигнал (SYN).
8	ENPAR		Разрешение 32 бита паритета для канала 1 – разрешена передача 32-м битом бита паритета 0 – разрешена передача 32-м битом бита данных Запрещено сбрасывать этот бит в ноль в штатном режиме работы контроллера
9	ODD		Выбор четности или нечетности бита паритета 1 – бит паритета формируется как дополнение до нечетности (если сумма всех разрядов данных по модулю 2 равно нулю, бит паритета устанавливается в 1, в противном случае в 0) 0 – бит паритета формируется как дополнение до четности (если сумма всех разрядов данных по модулю 2 равна единице, бит паритета устанавливается в 1, в противном случае в 0)
10	-		
11	-		
12	INTEDR		Разрешение прерывания наличие данных в FIFO 1 – разрешено прерывание, если FIFO приема данных не пусто 0 – прерывание запрещено
13	INTEER		Разрешение прерывания ошибка приема 1 – разрешено прерывания при возникновении ошибки в скорости приема или во времени паузы 4T между сообщениями (для сброса ошибки необходимо сбросить канал битом CH_EN) 0 – прерывание запрещено
14	INTEFF		Разрешение прерывания FIFO полно 1 – разрешено прерывание при полном заполнении FIFO данных 0 – прерывание запрещено
15	INTEHF		Разрешение прерывания FIFO наполовину полно 1 – разрешено прерывание, если FIFO наполовину или более полно 0 – прерывание запрещено
23..16	DIV[7:0]		Делитель частоты ядра до 1 МГц Содержит значение, на которое необходимо поделить частоту ядра, чтобы получить 1 МГц. Значение частоты не может быть более 255 МГц
31:24			

19.5.1 Регистр состояния приемника STATUS

Таблица 143 – Биты регистра состояния приемника STATUS

Бит	Имя	Значение	Описание
0	DR		Бит наличия данных в FIFO 0 – FIFO пусто 1 – FIFO содержит данные
1	ERR		Бит ошибки 0 – нет ошибок 1 – возникла ошибка приема
2	FF		Бит полноты FIFO 0 – FIFO не полно 1 – FIFO полно
3	HF		Бит наполненности FIFO 0 – FIFO не наполнено до половины 1 – FIFO заполнено на половину и более
4	IRQ_DR		Запрос прерывания при наличии данных в FIFO 0 – нет запроса 1 – есть запрос
5	IRQ_ERR		Запрос прерывания при наличии ошибки 0 – нет запроса 1 – есть запрос
6	IRQ_FF		Запрос прерывания при полном заполнении FIFO 0 – нет запроса 1 – есть запрос
7	IRQ_HF		Запрос прерывания при заполнении FIFO наполовину или более 0 – нет запроса 1 – есть запрос
8	IRQ_RX_CH0		Запрос прерывания от канала 0 0 – нет запроса 1 – есть запрос
9	IRQ_RX_CH1		Запрос прерывания от канала 1 0 – нет запроса 1 – есть запрос
10	IRQ_RX_CH2		Запрос прерывания от канала 2 0 – нет запроса 1 – есть запрос
11	IRQ_RX_CH3		Запрос прерывания от канала 3 0 – нет запроса 1 – есть запрос
12	IRQ_RX_CH4		Запрос прерывания от канала 4 0 – нет запроса 1 – есть запрос
13	IRQ_RX_CH5		Запрос прерывания от канала 5 0 – нет запроса 1 – есть запрос
14	IRQ_RX_CH6		Запрос прерывания от канала 6 0 – нет запроса 1 – есть запрос
15	IRQ_RX_CH7		Запрос прерывания от канала 7 0 – нет запроса 1 – есть запрос
23..16	CHAN_WP		Указатель на запись данных
31..24	DATA_RP		Указатель на чтение данных

Все приемники имеют один вход запроса прерывания в контроллере прерываний. Дополнительная информация о конкретном источнике прерываний может быть получена из регистра состояний.

### 19.5.2 LABEL

Буфер меток, с которыми сравниваются первые восемь принимаемых бит, если установлен LB\_EN бит соответствующего канала. Размер буфера для каждого канала 256x1. Это означает что каждой из 256 кодовых комбинаций метки поставлен в соответствие один бит буфера. Процесс сравнения выполняется путем чтения соответствующего бита сразу после приема кода метки. К моменту приема всего сообщения бит из буфера должен быть получен и если его значение равно 1, это соответствует приему сообщения с заданной меткой.

### 19.5.3 DATA\_R

#### FIFO принимаемых данных

В FIFO помещаются 32-разрядные данные, принимаемые из соответствующего канала. Размер FIFO для каждого канала разный:

- канал 1 – 256x32;
- канал 2 – 256x32;
- канал 3 – 64x32;
- канал 4 – 64x32;
- канал 5 – 64x32;
- канал 6 – 64x32;
- канал 7 – 32x32;
- канал 8 – 32x32.

Наличие или отсутствие данных в FIFO контролируется битами статуса DR, HF, FF соответствующего канала. При чтении данных указатель чтения автоматически инкрементируется. При записи данных в FIFO после приема указатель записи инкрементируется. Все FIFO каналов приема физически размещены в одном модуле синхронной статической памяти. Буфер каждого канала имеет смещение относительно начала памяти в соответствии с размером предыдущих буферов. Об этом нужно помнить при прямом доступе в память приемников.

#### Доступ к памяти

Все буферы приемников и меток расположены физически в одном модуле памяти. Поэтому при попытке обращения к буферу со стороны приемников и процессора может возникать конфликт. Он разрешается следующим образом. Если есть запрос от процессора, он имеет самый высокий приоритет и обслуживается первым. Для выбора обслуживаемого канала используется 3-х битовый счетчик, который циклически выбирает канала для обслуживания. Таким образом, в течение 8-ми тактов любой канала может быть обслужен. Запрос на доступ к памяти канала выставляет при приеме сообщения или при приеме метки. После приема сообщения у канала есть 32 такта времени ожидания на обслуживание. При приеме метки остается 22 такта ожидания. Под тактом понимается время приема одного бита сообщения, т.е. 100 КГц или 12,5 КГц. Поскольку частота шины при этом должна быть как минимум в 10 раз выше (для 100 КГц), то при работе на 1 МГц частоты шины мы имеем 220 тактов частоты шины, чтобы обслужить максимум 16 запросов

(8 сообщений и 8 меток). Приостановку обслуживания запросов может вызывать доступ к памяти со стороны процессора. Однако один цикл чтения памяти процессором занимает минимум 3 такта (в действительности больше) из которых только один есть обращение к памяти, а два других могут быть использованы для обслуживания каналов. Задержку обслуживания может вызвать обращение к памяти по записи со стороны процессора. Запись может выполняться каждый такт шины. Это нужно учитывать при работе с интерфейсом и не выполнять непрерывно записи в буфер приемника большого блока данных.

## 19.6 Описание регистров передатчика

**Таблица 144 – Регистры передатчика**

Базовый адрес	номер	Название	Состояние после сброса	Описание
0x8000_3000		ARINC429T0		Контроллер интерфейса передатчиков ARINC429
+0				Канал 0
	0	CONTROL		Регистр управления передатчика
	1	-		
	2	DATA_T		Регистр передаваемых данных
	3	STATUS		Регистр состояния передатчика
+4				Канал 1
+8				Канал 2
+12				Канал 3
0x8000_3400	0x000-0x1FF	Память передатчиков		Прямой доступ в память передатчиков
	0x000	TX0_base		Базовый адрес передатчика 0
	0x100	TX1_base		Базовый адрес передатчика 1
	0x140	TX2_base		Базовый адрес передатчика 2
	0x180	TX3_base		Базовый адрес передатчика 3
	0x1c0	FREE_base		Неиспользуемая область памяти

### 19.6.1 Регистр управления передатчиком CONTROL

**Таблица 145 – Биты регистра управления передатчиком CONTROL**

Бит	Имя	Значение	Описание
0	CH_EN		Разрешение работы канала 1 – передача по каналу разрешена 0 – канал передачи находится в состоянии сброса
1	CLK		Скорость передачи данных 1 – частота передаваемых данных = опорная частота/80 (12,5 кГц) 0 – частота передаваемых данных = опорная частота/10 (100 кГц)
2	EN_PAR		Разрешение 32 бита паритета 1 – разрешена передача 32-м битом бита паритета 0 – разрешена передача 32-м битом бита данных

3	ODD		Выбор четности или нечетности бита паритета 1 – бит паритета формируется как дополнение до нечетности (если сумма всех разрядов данных по модулю 2 равно нулю, бит паритета устанавливается в 1, в противном случае в 0) 0 – бит паритета формируется как дополнение до четности (если сумма всех разрядов данных по модулю 2 равна единице, бит паритета устанавливается в 1, в противном случае в 0)
4	INTE_TXR		Разрешение прерывания при опустошении буфера FIFO 1 – разрешено прерывание FIFO передачи данных пусто 0 – прерывание запрещено
5	INTE_FFT		Разрешение прерывания при полном заполнении буфера FIFO 1 – разрешено прерывание при полном заполнении FIFO данных 0 – прерывание запрещено
6	INTE_HFT		Разрешение прерывания при заполнении наполовину или менее буфера FIFO 1 – разрешено прерывание FIFO наполовину или менее полно 0 – прерывание запрещено
7	ENSYNC		Разрешение работы выходов передатчика в режиме данных и синхросигнала 1 – разрешено 0 – запрещено При установленном бите ENSYNC для соответствующего канала выход OUT_A работает как данные (D), выход OUT_B как синхросигнал (SYN)
15..8	DIV[7:0]		Делитель частоты ядра до 1 МГц Содержит значение, на которое необходимо поделить частоту ядра, чтобы получить 1 МГц

### 19.6.2 Регистр состояния передатчика STATUS

**Таблица 146 – Биты регистра состояния передатчика STATUS**

Бит	Имя	Значение	Описание
0	TX_R		Флаг наличия данных в FIFO 1 – FIFO пусто 0 – FIFO содержит данные
1	BUSY		1 – передатчик выполняет передачу данных 0 – передатчик в ожидании новых данных
2	FFT		Флаг полноты FIFO 1 – FIFO полно 0 – FIFO не полно
3	HFT		Флаг наполненности FIFO канала 1 – FIFO наполнено до половины или менее 0 – FIFO наполнено более чем на половину
4	IRQ_TXR		Запрос прерывания если FIFO пусто. 0 – нет запроса 1 – есть запрос

5	-		
6	IRQ_FFТ		Запрос прерывания при полном заполнении FIFO. 0 – нет запроса 1 – есть запрос
7	IRQ_HF		Запрос прерывания при заполнении FIFO наполовину или менее. 0 – нет запроса 1 – есть запрос
8	IRQ_TX_CH0		Запрос прерывания от канала 0 0 – нет запроса 1 – есть запрос
9	IRQ_TX_CH1		Запрос прерывания от канала 1 0 – нет запроса 1 – есть запрос
10	IRQ_TX_CH2		Запрос прерывания от канала 2 0 – нет запроса 1 – есть запрос
11	IRQ_TX_CH3		Запрос прерывания от канала 3 0 – нет запроса 1 – есть запрос
15..12	-		
23..16	DATA_WP		Значение указателя на запись данных
31..24	CHAN_RP		Значение указателя на чтение данных

Все передатчики имеют один вход запроса прерывания в контроллере прерываний. Дополнительная информация о конкретном источнике прерываний может быть получена из регистра состояний.

### **19.6.3 DATA\_T**

#### *FIFO передаваемых данных*

FIFO может содержать данные объемом:

- 256x32 для передачи по каналу 1;
- 64x32 для передачи по каналу 2;
- 64x32 для передачи по каналу 3;
- 64x32 для передачи по каналу 4.

Наличие или отсутствие данных в FIFO контролируется битами статуса TX\_R, HFT, FFT.

FIFO доступно по чтению и записи, однако только чтение вызывает инкремент указателя чтения.

Все FIFO каналов передачи физически размещены в одном модуле синхронной статической памяти. Буфер каждого канала имеет смещение относительно начала памяти в соответствии с размером предыдущих буферов. Об этом нужно помнить при прямом доступе в память передатчиков.

## 20 Контроллер МКПД (MIL\_STD)

В микроконтроллере имеется два независимых контроллера интерфейса Мультиплексного Канала Передачи Данных (МКПД) по ГОСТ Р 52070-2003 (далее МКПД). Каждый из которых, содержит необходимую логику и память для обработки и хранения командных слов и слов данных одного полного сообщения МКПД. Каждый контроллер содержит два канала для приема/передачи сообщений МКПД: основной и резервный. В один момент времени может работать только один из каналов основной или резервный. Одновременная работа двух каналов не предусмотрена. Контроллер может работать как в режиме контроллера шины, так и в режиме оконечного устройства. Для хранения входящих и исходящих командных и статусных слов, а также команд управления используются 16-разрядные регистры. Для хранения данных используется шестнадцатиразрядная память, в которой данные хранятся в области памяти соответствующей подадресу командного слова. В каждом подадресе можно хранить только одно полное сообщение МКПД (до 32-х слов данных). При передаче сообщения данные в память можно заносить как на «лету», так и до начала передачи. При приеме сообщения, данные можно считывать из памяти, как на «лету», так и после установки флага VALMESS.

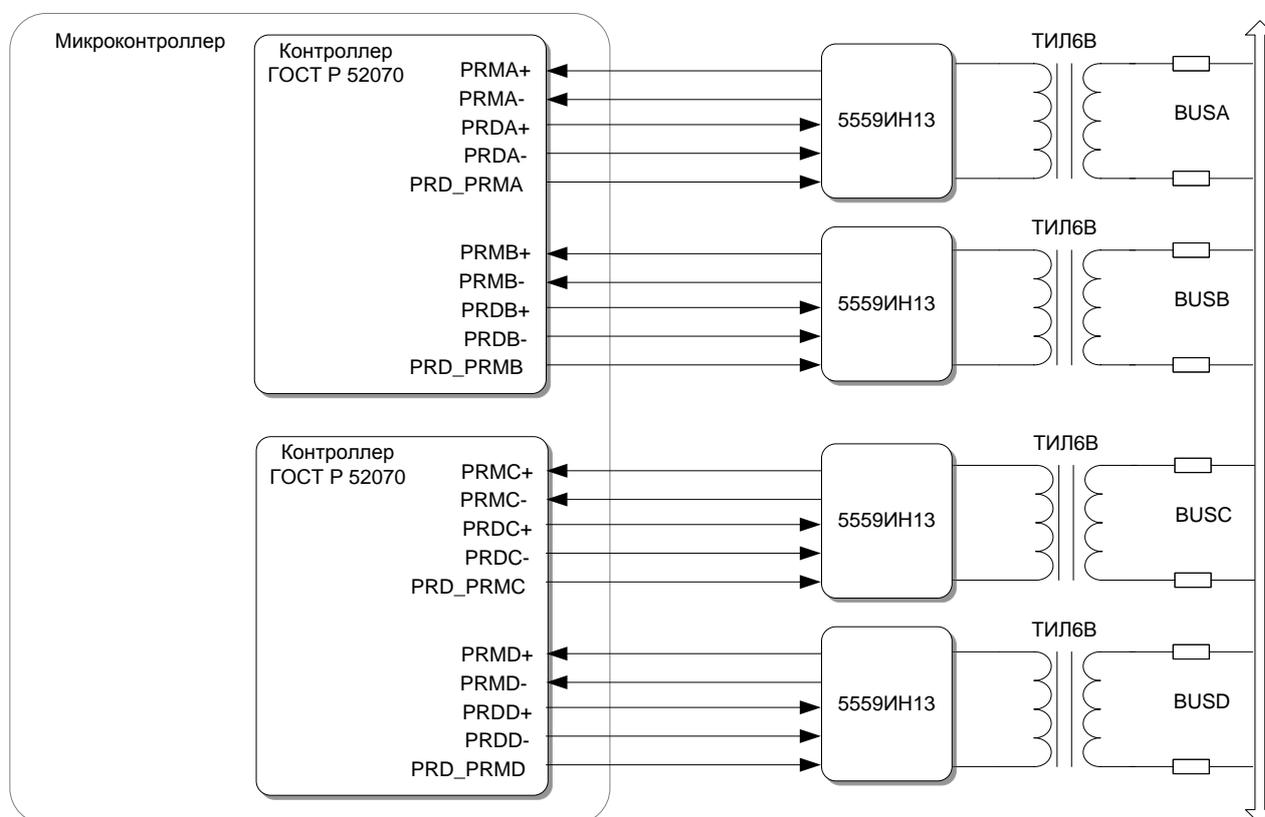


Рисунок 66

Особенности:

- поддержка основных (формат 1 – формат 6) и групповых (формат 7 – формат 10) форматов сообщений;
- поддержка режимов работы: контроллер шины, оконечное устройство, монитор;
- скорость передачи данных 1 Мбит/с в полудуплексном режиме;

- поддержка двух каналов связи: основного и резервного;
- память принимаемых данных 1Кх16;
- память передаваемых данных 1Кх16;
- возможность формирования прерываний при успешном приеме и при возникновении ошибок на шине;
- маскирование прерываний;
- второй контроллер МКПД доступен только в режиме работы с 16-разрядной внешней шиной данных.

## **20.1 Режимы работы**

Контроллер поддерживает три режима работы: контроллера шина (КШ), оконечного устройства (ОУ) и неадресуемого монитора (М).

### **20.1.1 Контроллер шины**

В этом режиме контроллер передает команды в магистраль, участвует в пересылке слов данных, принимает и контролирует ответную информацию о состоянии ОУ. Помимо этого КШ реализует все команды управления. Для того чтобы реализовать передачу командного слова в магистраль используется регистр CommandWord1. А для сообщений формата 3 и 8 помимо этого применяется регистр CommandWord2. Ответная информация о состоянии ОУ после приема из магистрали хранится в регистре StatusWord1. А для сообщений формата 3 и 8 помимо этого применяется регистр StatusWord2. Для передачи и приема слов данных команд управления (КУ), форматы сообщений 5, 6 и 10, применяется регистр ModeData. Выбор этого режима работы осуществляется в регистре CONTROL установкой бита BCMODE и сбросом RTMODE.

### **20.1.2 Оконечное устройство**

В этом режиме контроллер осуществляет проверку достоверности командных слов, поступающих к нему от КШ. Командное слово считается достоверным, если не возникло ошибок в магистрали при его приеме, или если поле «Адрес ОУ» соответствует коду собственного адреса ОУ или коду 11111 (групповая команда). Если командное слово определено как достоверное, ОУ посылает в линию ответное слово (ОС) и в зависимости от поля «Прием/Передача» принимает или передает число данных, соответствующее полю «Число СД/Код КУ». Если же происходит прием от КШ команды управления, ОУ реагирует в соответствии с форматами сообщений команд управления. Принятое из магистрали командное слово помещается в регистр CommandWord1, а для сообщений формата 3 и 8 принятое второе командное слово помещается в регистр CommandWord2. Ответное слово ОУ для передачи в магистраль помещается в регистр StatusWord1. Помимо этого, для сообщения формата 3, этот регистр содержит принятое ответное слово от другого ОУ. Для передачи и приема слов данных команд управления, форматы сообщений 5, 6 и 10, применяется регистр ModeData. Выбор этого режима работы осуществляется в регистре CONTROL установкой бита RTMODE и сбросом BCMODE.

### 20.1.3 Монитор

В этом режиме осуществляется прослушивание магистрали и отбор необходимой информации для проведения: технического обслуживания, регистрации эксплуатационных параметров, анализа решаемых задач или обеспечения информацией резервного КШ. Монитор пассивно прослушивает выбранную шину и захватывает весь трафик на шине, но никогда не передает информацию на шину. Принятое из магистрали командное слово помещается в регистр CommandWord1, а для сообщений формата 3 и 8 принятое второе командное слово помещается в регистр CommandWord2. Ответное слово ОУ принятое из магистрали помещается в регистр StatusWord1. А для сообщений формата 3 и 8 помимо этого применяется регистр StatusWord2. Для приема слов данных команд управления, форматы сообщений 5, 6 и 10, применяется регистр ModeData. Выбор этого режима работы осуществляется в регистре CONTROL установкой битов RTMODE и BCMODE. Для быстрой расшифровки сообщений можно применить регистр MSG. Каждому формату сообщения на магистрали соответствует определенный код в этом регистре.

### 20.2 Форматы сообщений

Сообщения, передаваемые по информационной магистрали, имеют формат, соответствующий форматам основных или групповых сообщений. Любые другие типы сообщений, не соответствующие ГОСТ Р 52070-2003 не поддерживаются.

Форматы основных сообщений, приведенные ниже на рисунке, используются для передачи информации предназначенной одному ОУ и предусматривают выдачу ОС. В данном случае КС – командное слово, СД – слово данных, ОС – ответное слово. Времена  $t_1$  и  $t_2$  формируются аппаратно и не могут быть изменены программно. Пауза  $t_2$  между сообщениями, формируемая КШ, не менее 4 мкс. А пауза перед передачей ОС, формируемая ОУ, в пределах от 4 до 12 мкс. Если после ожидания 14 мкс так и не поступило ОС от ОУ, фиксируется отсутствие ОС от ОУ и формируется соответствующий признак ошибки.

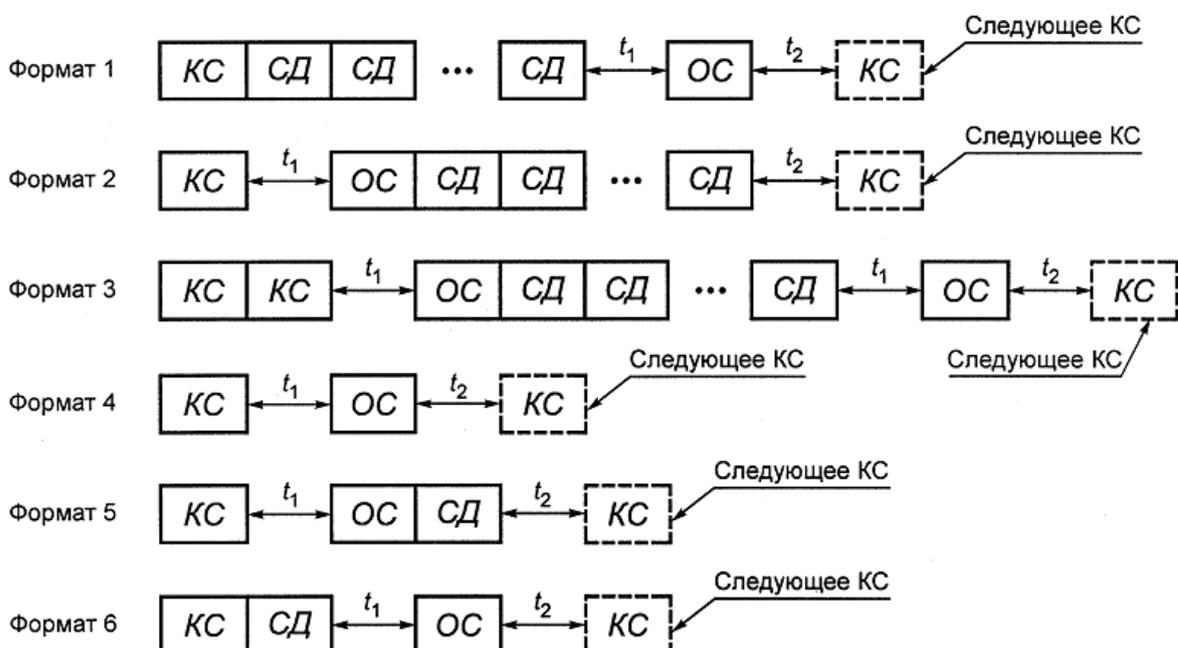


Рисунок 67 – Форматы 1 ÷ 6

Время непрерывной передачи данных в линию не превышает 660 мкс, что соответствует командному слову и 32-м словам данных.

Формат 1 – передача данных от КШ к ОУ.

Формат 2 – передача данных от ОУ к КШ.

Формат 3 – передача данных от ОУ к ОУ.

Формат 4 – передача КУ.

Формат 5 – передача КУ и прием СД от ОУ.

Формат 6 – передача КУ и СД окончному устройству.

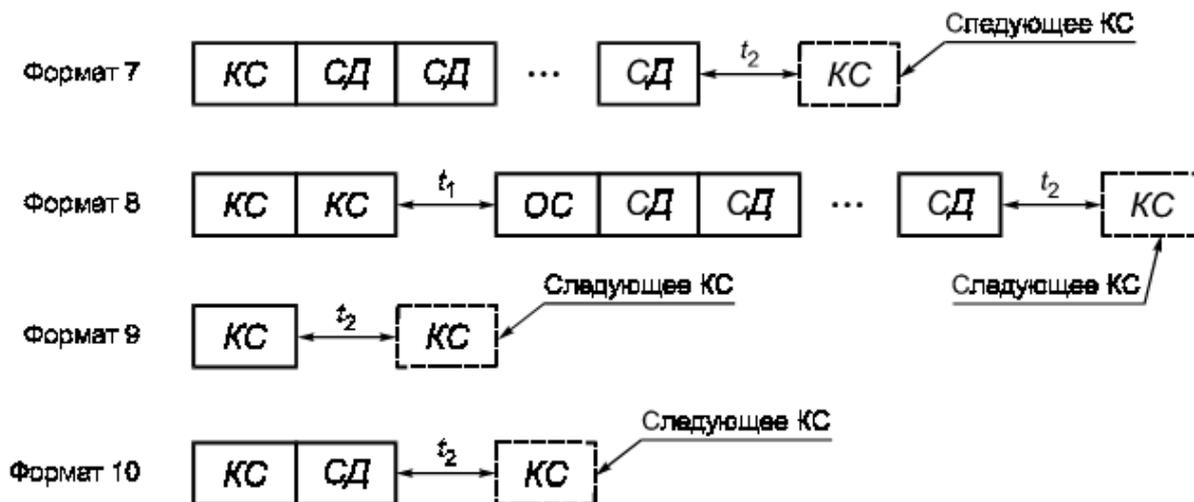
Групповые сообщения, приведенные ниже (Рисунок 68), начинающиеся с передачи КШ групповой команды с кодом адреса 11111, используются для передачи информации одновременно нескольким ОУ без выдачи ими ОС.

Формат 7 – передача данных (в групповом сообщении) от КШ к оконечным устройствам.

Формат 8 – передача данных (в групповом сообщении) от оконечного устройства к оконечным устройствам.

Формат 9 – передача групповой команды управления.

Формат 10 – передача групповой команды управления со словом данных.



**Рисунок 68 – Форматы 7 ÷ 10**

Если ОУ в формате сообщения ОУ-ОУ получило достоверное командное слово на прием информации, то первое СД должно быть им принято через паузу не более  $(57 \pm 3)$  мкс, в противном случае формируется соответствующий признак ошибки.

### 20.3 Формат слов

Каждое слово начинается с сигнала пословной синхронизации (с синхросигнала) и имеет 17 информационных разрядов, включая разряд контроля по четности. Форматы слов приведены ниже (Рисунок 69).

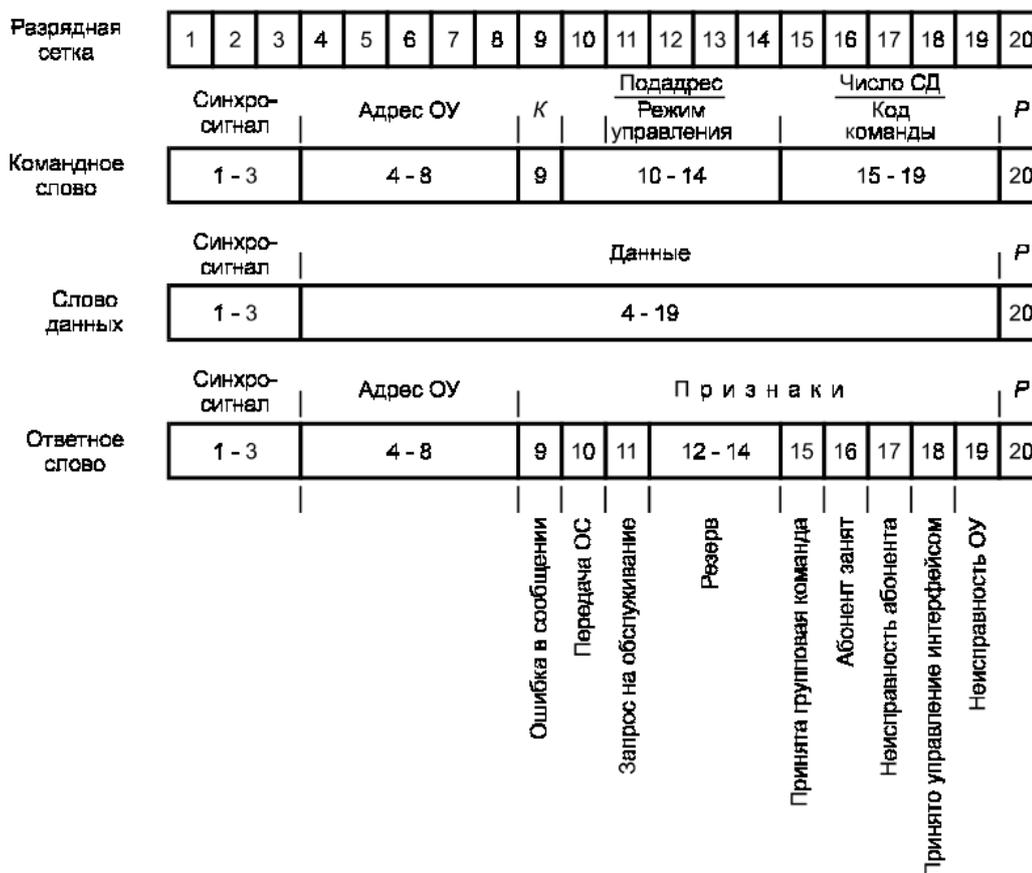


Рисунок 69 – Форматы слов

Командное слово содержит:

- синхросигнал;
- поле «Адрес ОУ»;
- разряд «Прием/Передача»;
- поле «Подадрес/Режим управления»;
- поле «Число СД/Код КУ»;
- разряд контроля по четности.

Синхросигнал имеет длительность, составляющую три интервала времени передачи одного двоичного разряда. Полярность первой половины сигнала положительная, а вторая отрицательная.

Адрес ОУ содержит код адреса из диапазона кодов 00000 – 11110, которому предназначено КС. КС с кодом адреса 11111 называется групповой командой, а сообщение, содержащее групповую команду – групповым.

Разряд «Прием/Передача» указывает на действие, которое должно выполнить ОУ (принимать или передавать СД). Логический ноль означает, что ОУ должно принимать СД, а логическая единица – передавать СД.

Поле «Подадрес/Режим управления» содержит код подадреса ОУ или код признака режима управления 00000 или 11111.

Поле «Число СД/Код КУ» содержит код числа слов данных, которые должны быть переданы или приняты ОУ в связи с приемом адресованного ему КС, или код КУ. В одном сообщении может быть передано не более 32 СД. Числовое значение двоичных кодов, обозначающих число СД, соответствует их десятичным эквивалентам, за исключением кода 00000, который соответствует числу 32.

Разряд контроля по четности используется для контроля по четности предшествующих ему 16 разрядов КС. Разряд принимает такое значение, чтобы сумма значений всех 17 информационных разрядов слова (включая контрольный разряд) была нечетной.

Слово данных содержит:

- синхросигнал;
- данные;
- разряд контроля по четности.

Синхросигнал имеет длительность, составляющую три интервала времени передачи одного двоичного разряда. Полярность первой половины сигнала отрицательная, а вторая положительная.

Поле данных содержит передаваемые данные, а разряд контроля по четности формируется так же, как в командном слове.

Ответное слово содержит:

- синхросигнал;
- поле «Адрес ОУ»;
- поле разрядов признаков состояния: ошибка в сообщении, передача ОС, запрос на обслуживание, принята групповая команда, абонент занят, неисправность абонента, принято управление интерфейсом, неисправность ОУ;
- разряд контроля по четности.

Синхросигнал аналогичен синхросигналу КС. Поле «Адрес ОУ» содержит собственный адрес ОУ. Поле разрядов признаков состояния ОУ отображает текущее состояние ОУ. Разряд контроля по четности формируется так же, как в командном слове.

## 20.4 Структурная схема в режиме КШ

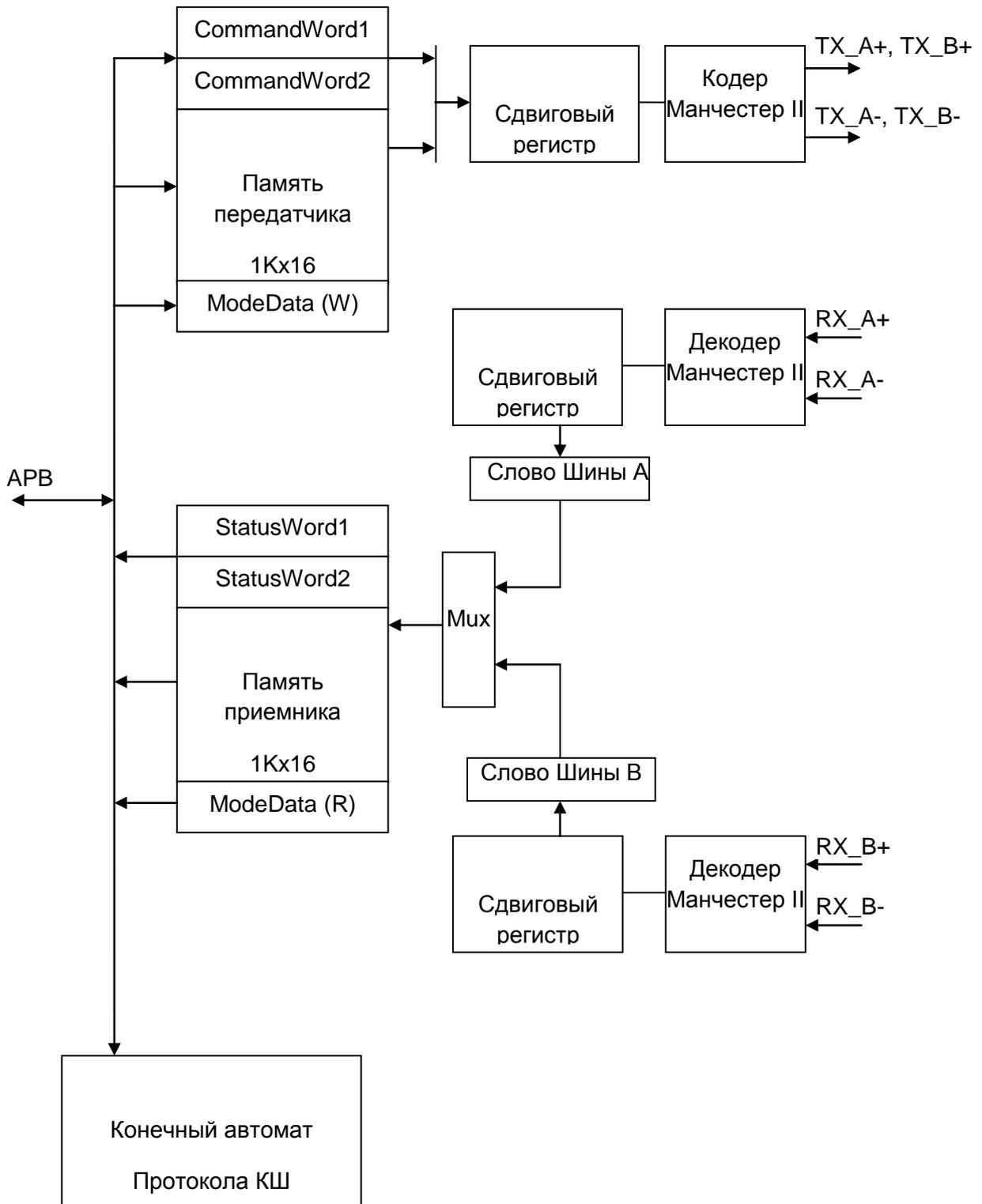


Рисунок 70 – Структурная схема в режиме КШ

## 20.5 Структурная схема в режиме ОУ

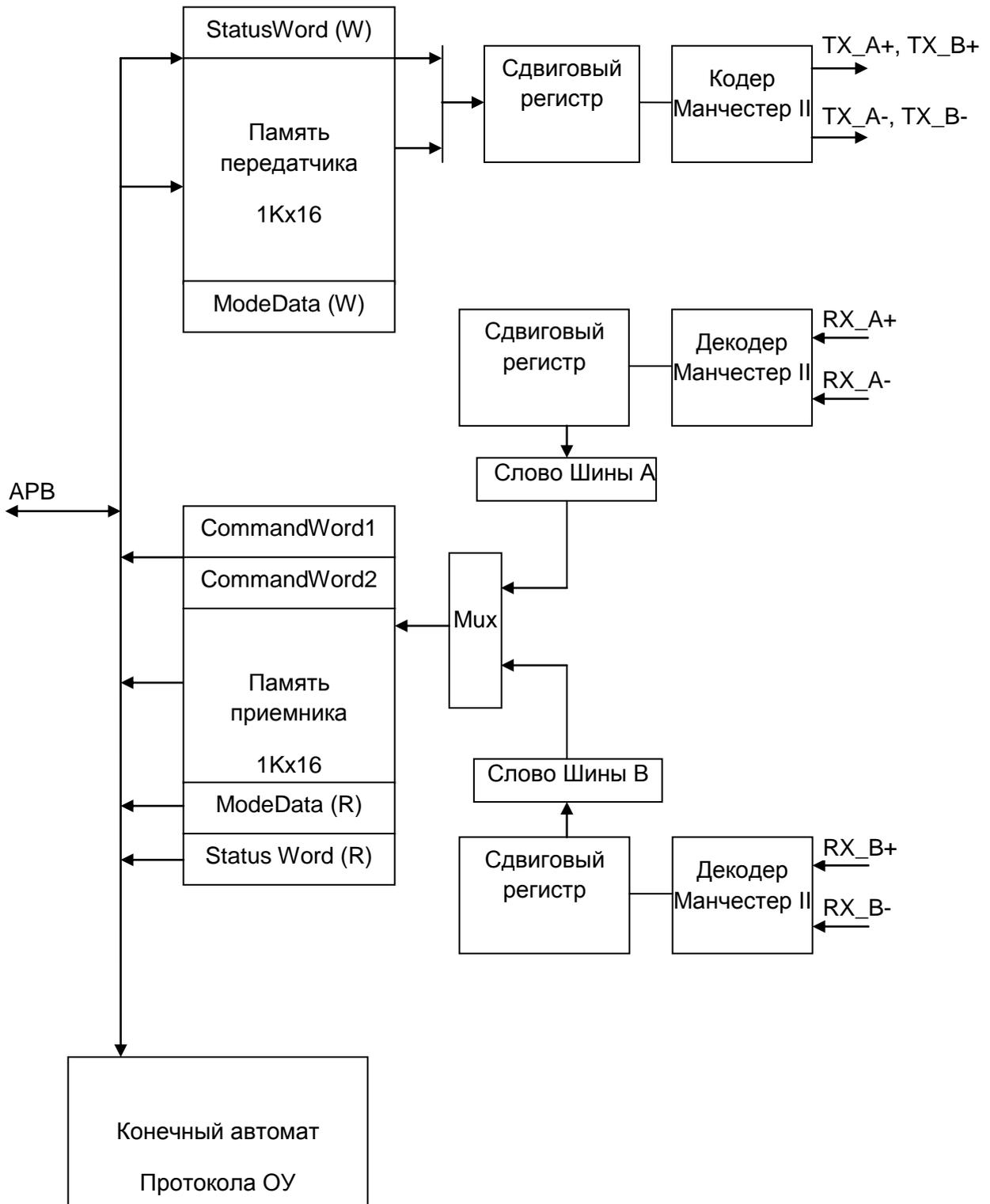


Рисунок 71 – Структурная схема в режиме ОУ

## 20.6 Структурная схема в режиме М

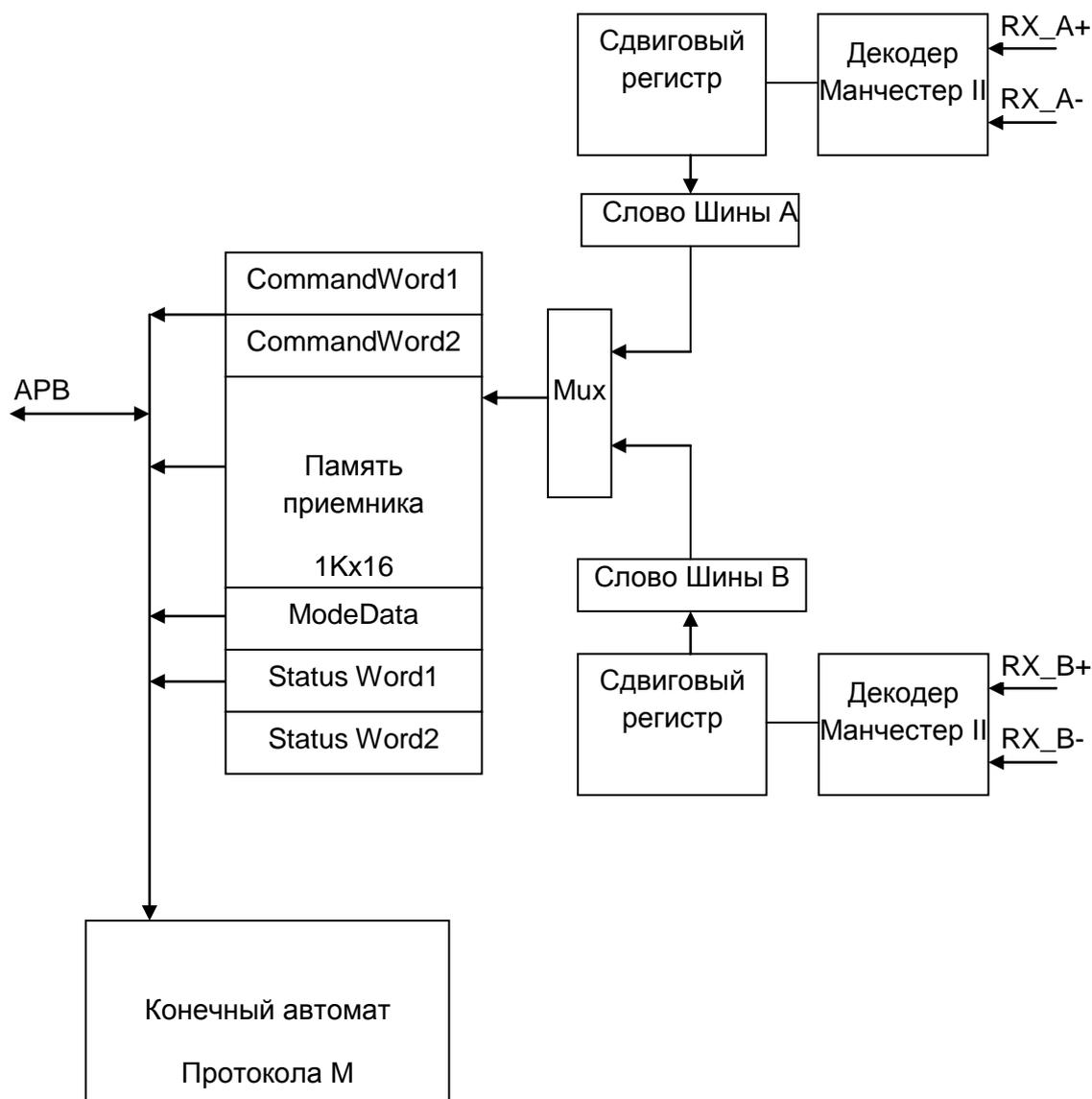


Рисунок 72 – Структурная схема в режиме М

## 20.7 Инициализация

Перед началом работы с контроллером в первую очередь необходимо сбросить контроллер, чтобы очистить все регистры сообщений. Это достигается установкой бита MR регистра CONTROL в логическую единицу. Затем бит необходимо сбросить в нуль. Далее нужно задать в регистре CONTROL значение делителя частоты DIV таким образом, чтобы при делении частоты ядра HCLK на это значение получить опорную частоту блока контроллера 1 МГц. После этого с помощью бит RTMODE и BCMODE выбирается соответствующий режим работы ОУ или КШ. Одновременная установка двух этих бит запрещена.

Для того чтобы выбрать какой канал будет использован для передачи данных основной или резервный, устанавливается соответствующий бит (TRA – основной канал, TRB – резервный канал). В режиме КШ командные слова будут передаваться

только по тому каналу, который выбран. В режиме ОУ необходимо установить оба бита, так как ОУ не может выбирать, по какому каналу ему передавать СД и ОС, и поэтому их передача происходит по тому каналу, по которому было принято КС.

Для режима ОУ в битах RTA4 – RTA0 регистра CONTROL задается адрес ОУ, который должен соответствовать адресу в поле «Адрес ОУ» командного слова, если идет обращение к этому ОУ.

*Пример инициализации ОУ*

```
*((volatile unsigned int *) (0x80006400))=0x00000001; //установка бита MR=1
*((volatile unsigned int *) (0x80006400))= 0x00014078;
//RTMODE=1, TRB=TRA=1, RTA=1, DIV=40
```

*Пример инициализации КШ*

```
*((volatile unsigned int *) (0x80006400))=0x00000001; //установка бита MR=1
*((volatile unsigned int *) (0x80006400))= 0x00014014;
//BCMODE=1, TRA=1, TRB=0, RTA=0, DIV=40
```

В обоих случаях значения делителя частоты DIV = 40, что соответствует частоте ядра 40 МГц, и для получения опорной частоты контроллера необходимо 40 МГц/DIV = 1 МГц.

## **20.8 Прием и передача в режиме ОУ**

Для того чтобы настроить контроллер в режиме ОУ, необходимо выполнить все пункты описанные в параграфе Инициализация. После этого необходимо задать ответное слово для КШ с помощью регистра StatusWord1. В режиме ОУ регистр по записи содержит предназначенное для передачи КШ ответное слово, а по чтению содержит ответное слово, полученное от передающего ОУ в транзакции ОУ-ОУ.

Дополнение режима ОУ

В режиме работы ОУ добавлена возможность задания адреса ОУ из внешнего источника. Для этого за каждым контроллером закреплен набор внешних контактов, которые определяют адрес ОУ. Данные внешние контакты анализируются и используются только в случае, когда интерфейс сконфигурирован для работы в режиме ОУ, а также бит 23 регистра управления установлен в 1. Биты 10..6 регистра управления в этом случае доступны только по чтению и отражают заданный адрес. После включения функции использования внешнего адреса, данный адрес непрерывно анализируется на наличие ошибки четности (количество единиц в пяти битах адреса и одном бите четности всегда должно быть нечетным).

Если возникает ошибка то:

- флаг ошибки можно прочитать в регистре ERROR;
- возможна генерация прерывания, если установлен бит разрешения в INTEN;
- блокируется работа автомата протокола ОУ.

Адрес ОУ должен быть установлен до момента включения контроллера в режиме ОУ. В противном случае должны быть предусмотрены меры, исключаящие случайную генерацию ошибки при смене адреса.

*Пример записи ответного слова ОУ*

```
*((volatile unsigned int *) (0x80006406))=0x00000800;
```

В данном случае в регистр заносятся только старшие 5 разрядов, соответствующие адресу ОУ. Остальные разряды признаки состояния ОУ можно оставить в нуле. Но в процессе работы может возникнуть необходимость изменять эти биты. Для этого необходимо программно устанавливать и сбрасывать эти биты, так как аппаратно они не изменяются.

Для того чтобы обеспечить формат сообщения 5, необходимо задавать слово данных, передаваемое КШ в команде управления. Для этих целей используется регистр ModeData.

*Пример записи слова данных команды управления*

```
*((volatile unsigned int *) (0x80006405))=0x000055AA;
```

После того как проведена инициализация, заданы ответное слово и слово данных команды управления, ОУ сразу готово к работе и может отвечать на все возможные форматы сообщений.

Так как в процессе работы ОУ в каждый момент времени требуется передача определенных СД и СД команды управления, то программно необходимо обновлять те области памяти, которые содержат эти данные. Если эти области не обновляются, при запросе данных КШ будут переданы те данные, которые были последний раз записаны в эти области памяти. Поэтому при написании программы следует помнить и обновлять данные и слова данных команды управления.

Для хранения СД передатчика применяется адресное пространство 0x000-0x3FF (относительно базового адреса периферийного блока). Данные шестнадцатиразрядные, старшие 16 бит всегда равны нулю при чтении и игнорируются при записи.

*Пример инициализации данных для подадреса 1*

```
addon=0x20;  
for(i=1;i<=32;i++)  
{  
*((volatile unsigned int *) (0x80006000+addon))=i;  
addon=addon+1;  
}
```

Из примера становится ясно, что стартовый адрес памяти СД для подадреса 1 – 0x20, для последующих подадресов: n\*0x20, где n- номер подадреса (n=1-31).

При приеме СД или слова данных команды управления, признаком обновления их значений является флаг VALMESS. После того как флаг установлен можно считать новые данные или слово данных команды управления. Но следует учитывать то, что этот флаг автоматически сбрасывается через 4 мкс, после его установки, поэтому желательно применять прерывания по установке сигнала VALMESS.

*Пример чтения слова данных команды управления*

```
i=*((volatile unsigned int *) (0x80006405));
```

*В переменную i будет прочитано значение слова данных команды управления.*

*Пример чтения данных для подадреса 1*

```

addon=0x20;
for(i=1;i<=32;i++)
{
    mas[i]=*((volatile unsigned int*)(0x80006800+addon));
    addon=addon+1;
}
    
```

Для упрощения декодирования команд управления КШ в режиме ОУ доступен регистр кодов MSG полученных сообщений. Каждому формату сообщения на магистрали соответствует определенный код в этом регистре. Чтение и последующая дешифрация этого кода упрощает процедуру декодирования сообщения и уменьшает время обработки сообщений. При использовании регистра экономится время на чтение двух командных регистров и разбор значений бит этих регистров.

## **20.9 Прием и передача в режиме КШ**

В отличие от режима ОУ, в режиме КШ необходимо задавать не ответное слово, а командное слово или два командных слова в режиме работы ОУ-ОУ. Помимо этого нужно инициировать процедуру приема или передачи данных установкой бита BCSTART. После завершения транзакции на шине этот бит автоматически сбрасывается в ноль. Поэтому для инициирования новой транзакции нужно повторно устанавливать этот бит.

*Пример записи командных слов и бита BCSTART*

```

*((volatile unsigned int*)(0x80006403))=0x00000820; //Командное слово 1
*((volatile unsigned int*)(0x80006404))=0x00000000; //Командное слово 2
*((volatile unsigned int*)(START_ADDR_APB+0x400))=0x00014016; //Регистр
управления
    
```

Как видно из примера, в командном слове 1 задается код слов данных 00000, что соответствует 32 СД. Данные будут передаваться от контроллера шины оконечному устройству с адресом 1 из подадреса 1. Второе командное слово задается равным нулю и никак не влияет на транзакцию. В регистре управления устанавливается бит BCMODE, что соответствует режиму работы КШ, а также устанавливается бит BCSTART, что иницирует начало транзакции, выбирается канал А для передачи (TRA=1), а также устанавливается делитель частоты 40, что соответствует частоте работы ядра 40 МГц.

Для того чтобы инициировать прием в этом примере, необходимо только установить бит 10 равным единице в командном слове 1.

Если транзакция завершена успешно (признак VALMESS установился в единицу), полученные СД или слово данных команды управления могут быть прочитаны. В противном случае устанавливается один из флагов ошибки. Сброс этих флагов осуществляется установкой битом MR или инициированием новой транзакции битом BCSTART.

## 20.10 Прерывания

Для уменьшения потерь времени программы на опрос флагов контроллера, введено одно прерывание, генерируемое при установке любого из флагов контроллера. Прерывание может генерировать установка одного из четырех флагов:

- флаг ошибки;
- флаг успешного завершения транзакции в канале;
- флаг приема достоверного КС, ОС или слова данных команды управления;
- флаг неактивности контроллера.

Каждый из флагов может быть маскирован битами разрешения прерывания по какому-либо флагу.

## 20.11 Описание регистров

**Таблица 147 – Регистры контроллера МКПД (MIL\_STD)**

Базовый адрес	Смещение	Название	Состояние после сброса	Описание
0x8000_6000		MIL-STD-1553B0		
0x8000_7000		MIL-STD-1553B1		
	0x000-0x3FF	DATA		Память передаваемых СД
	0x400	CONTROL		Регистр управление контроллером
	0x401	STATUS		Регистр состояния контроллера
	0x402	ERROR		Регистр ошибок контроллера
	0x403	CommandWord1		Регистр командного слова 1
	0x404	CommandWord2		Регистр командного слова 2
	0x405	ModeData		Слово данных команды управления
	0x406	StatusWord1		Регистр ответного слова 1
	0x407	StatusWord2		Регистр ответного слова 2
	0x408	INTEN		Регистр разрешения прерываний
	0x409	MSG		Регистр декодирования сообщений
	0x800-0xBFF	DATA		Память принимаемых СД

20.11.1 Регистр управления CONTROL

Таблица 148 – Биты регистра управления CONTROL

Бит	Имя	Значение	Описание
31..26			Зарезервировано
25	SYNC_DM		Выбор режима детектирования синхросигнала: 0 – режим 0 (версия 1986BE1Т) 1 – режим 1 (версия TS201)
24	DIV7		Старший бит делителя частоты
23	EN_XRTA		Когда равен 1 разрешает использование внешнего адреса для контроллера работающего в режиме оконечного устройства
22	ENFILTER3		
21	AUTOTUNE		Бит автоматической подстройки середины битовых интервалов (с ревизии 3) 0 – автоподстройка разрешена 1 – автоподстройка запрещена
20	ENFILTER		Бит разрешения фильтрации потока NRZ (с ревизии 3) 1 – фильтрация разрешена 0 – фильтрация запрещена В случае применения драйверов с некорректной скважностью и длительностью импульсов NRZ кода, таких как 5559ИН67Т, необходимо устанавливать бит в единицу для корректного приема. В этом случае контроль длительностей импульсов NRZ не осуществляется.
19	INVTR		Разрешение инверсии сигналов (с ревизии 3) PRD_PRMA, PRD_PRMB, PRD_PRMC, PRD_PRMD 1 – инверсия 0 – прямой выход
18	RERR		Сброс ошибок в режиме ОУ 1 – ошибки могут быть сброшены только битом MR 0 – сброс ошибок происходит автоматически, после установки бита IDLE
17..11	DIV6-DIV0		Делитель частоты периферийной шины до 1 МГц Содержит значение, на которое необходимо поделить частоту SCLK, чтобы получить МГц. Частота SCLK обязательно должна быть не более 255 МГц и не менее 8.
10..6	RTA4-RTA0		Адрес оконечного устройства Содержит адрес, который присвоен устройству, если контроллер работает в режиме оконечного устройства RTMODE=1; BCMODE=0 Если бит 23 установлен, данные биты недоступны по записи и всегда используется адрес задаваемый внешним источником. Адрес доступен только по чтению.
5	TRB		Блокировка передатчика резервного канала. 1 – передатчик разблокирован 0 – передатчик заблокирован
4	TRA		Блокировка передатчика основного канала. 1 – передатчик разблокирован 0 – передатчик заблокирован
3..2	RTMODE BCMODE		Выбор режима работы контроллера 10 – режим оконечного устройства 01 – режим контроллера шины 11 – режим неадресуемого монитора

1	BCSTART		Иницирует передачу сообщения в канал в режиме КШ. 1 – старт сообщения. 0 – стоп сообщения. Сбрасывается в ноль автоматически по завершению сообщения.
0	MR		Сброс контроллера. 1 – контроллер сбрасывается в исходное состояние 0 – разрешение работы контроллера

### 20.11.2 Регистр состояния STATUS

**Таблица 149 – Биты регистра состояния STATUS**

Бит	Имя	Значение	Описание
31..6	-		Зарезервировано
5	RCVB		Признак активности резервного канала 0 – канал неактивен 1 – канал активен
4	RCVA		Признак активности основного канала 0 – канал неактивен 1 – канал активен
3	ERR		Ошибка в сообщении. 0 – нет ошибок 1 – в последней транзакции возникла ошибка В режиме ОУ и М, если сброшен бит RERR, автоматически сбрасывается не менее чем через 4 мкс после установки.
2	VALMESS		Успешное завершение транзакции в канале. 0 – транзакция завершена с ошибкой или транзакции нет в канале 1 – транзакция завершена успешно В режиме ОУ и М автоматически сбрасывается не менее чем через 4 мкс после установки.
1	RFLAGN		Получено достоверное слово из канала. 0 – нет достоверных слов в канале 1 – в режиме КШ получено достоверное ответное слово 1 – в режиме ОУ или М получено достоверное командное слово, ответное слово или слово данных в команде управления Между сообщениями бит автоматически сбрасывается в ноль.
0	IDLE		Состояние контроллера. 1 – контроллер в неактивном состоянии 0 – контроллер в состоянии обмена сообщениями

20.11.3 Регистр ошибок ERROR

Таблица 150 – Биты регистра ошибок ERROR

Бит	Имя	Значение	Описание
31..8			Зарезервировано
7	RTAERR		Ошибка во внешнем адресе оконечного устройства 1 – есть ошибка 0 – нет ошибок
6	PROERR		Ошибка в протоколе. 1 – недопустимое слово обнаружено на шине во время обмена сообщениями 0 – нет ошибок
5	CONERR		Ошибка непрерывности сообщения. 1 – передача сообщения не непрерывная 0 – нет ошибок
4	GAPERR		Недопустимая активность на шине. 1 – обнаружена активность на шине в интервале 4 мкс после успешного завершения сообщения 0 – нет ошибок
3	CSYCERR/ SEQERR		Ошибка синхронизации команды в режиме КШ (CSYCERR). 1 – ожидался синхроимпульс команды, а получен синхроимпульс данных 0 – ошибок нет Ошибка после приема команды в режиме ОУ и М (SEQERR). 1 – обнаружена пауза после приема командного слова с битом 10 равным нулю или обнаружены слова данных после приема командного слова с битом 10 равным единице 0 – ошибок нет
2	DSYCERR/ SYNCERR		Ошибка синхронизации данных в режиме КШ (DSYCERR). 1 – ожидался синхроимпульс данных, а получен синхроимпульс команды 0 – ошибок нет Ошибка синхронизации в режиме ОУ и М (SYNCERR). 1 – ожидался синхроимпульс команды, а получен синхроимпульс данных или наоборот 0 – ошибок нет
1	MANERR		Ошибка декодирования NRZ кода. 1 – ошибка в количестве принятых бит или ошибка в бите контроля четности 0 – ошибок нет
0	NORCV		Ошибка приема. 1 – не получено ответное слово в интервале 14 мкс или не получены ожидаемые данные 0 – ошибок нет

### 20.11.4 Регистр команды 1 CommandWord1

**Таблица 151 – Биты регистра 1 CommandWord1**

Бит	Имя	Значение	Описание
31..16	-		Зарезервировано
15..11	Адрес ОУ		Адрес оконечного устройства, которому предназначено командное слово
10	Прием/передача		Бит приема/передачи 1 – режим работы ОУ-КШ 0 – режим работы КШ-ОУ
9..5	Подадрес / Режим управления		Содержит подадрес, по которому в памяти располагаются принимаемые или передаваемые СД. В случае передачи команды, содержит код 00000 или 11111
4..0	Число СД / Код команды		Содержит количество принимаемых или передаваемых слов данных. В случае передачи команды содержит код команды из Таблицы 1 ГОСТ Р52070-2003

*Примечание* – В режиме ОУ и М регистр доступен только на чтение, в режиме КШ только на запись.

### 20.11.5 Регистр команды 2 CommandWord2

**Таблица 152 – Биты регистра 2 CommandWord2**

Бит	Имя	Значение	Описание
31..16			Зарезервировано
15..11	Адрес ОУ		Адрес оконечного устройства, которому предназначено командное слово
10	Прием/передача		Бит приема/передачи 1 – режим работы ОУ-ОУ 0 – командное слово не используется
9..5	Подадрес		Содержит подадрес, по которому в памяти располагаются принимаемые или передаваемые СД
4..0	Число СД		Содержит количество принимаемых или передаваемых слов данных

*Примечание* – В режиме ОУ и М регистр доступен только на чтение и содержит второе командное слово транзакции ОУ-ОУ. В режиме КШ регистр доступен только на запись и используется для транзакции ОУ-ОУ, если установлен в единицу бит 10.

### 20.11.6 Слово данных команды управления ModeData

**Таблица 153 – Биты регистра ModeData**

Бит	Имя	Значение	Описание
31..16			Зарезервировано
15..0	ModeData		Содержит принятое или передаваемое слово данных в команде управления

### 20.11.7 Ответное слово 1 StatusWord1

**Таблица 154 – Биты регистра StatusWord1**

Бит	Имя	Значение	Описание
31..16			Зарезервировано
15..11	Адрес ОУ		Адрес ОУ, от которого принято ответное слово в режиме КШ. Адрес ОУ, которое передает ответное слово в режиме ОУ
10	Ошибка в сообщ.		Ошибка в сообщении
9	Пер.ОС		Передача ответного слова
8	Запр. на облс.		Запрос на обслуживание
7-5			Зарезервировано
4	Прин. ГК		Принята групповая команды
3	Абон. зан.		Абонент занят
2	Неисп. абон.		Неисправность абонента
1	Прин. упр. инт.		Принято управление интерфейсом
0	Неисп. ОУ		Неисправность ОУ

*Примечание* – В режиме ОУ регистр по записи содержит предназначенное для передачи КШ ответное слово, а по чтению содержит ОС, полученное от передающего ОУ в транзакции ОУ-ОУ. В режиме КШ и М регистр доступен только на чтение и содержит принятое от ОУ ответное слово.

### 20.11.8 Ответное слово 2 StatusWord2

**Таблица 155 – Биты регистра StatusWord2**

Бит	Имя	Значение	Описание
31..16			Зарезервировано
15..11	Адрес ОУ		Адрес ОУ, передающего данные в транзакции ОУ-ОУ
10	Ошибка в сообщ.		Ошибка в сообщении
9	Пер. ОС		Передача ответного слова
8	Запр. на облс.		Запрос на обслуживание
7-5			Зарезервировано
4	Прин. ГК		Принята групповая команды
3	Абон. зан.		Абонент занят
2	Неисп. абон.		Неисправность абонента
1	Прин. упр. инт.		Принято управление интерфейсом
0	Неисп. ОУ		Неисправность ОУ

*Примечание* – В режиме КШ и М регистр доступен только на чтение и содержит ОС передающего ОУ в транзакции ОУ-ОУ. В режиме ОУ регистр не используется.

### 20.11.9 Регистр разрешения прерываний INTEN

**Таблица 156 – Биты регистра разрешения прерываний INTEN**

Бит	Имя	Значение	Описание
31..5			Зарезервировано
4	RTAIE		Прерывание при ошибке в адресе оконечного устройства. 0 – прерывание маскировано 1 – прерывание разрешено
3	ERRIE		Прерывание при возникновении ошибки в сообщении. 0 – прерывание маскировано 1 – прерывание разрешено, это позволяет генерировать прерывание при возникновении ошибок в сообщении
2	VALMESSIE		Прерывание при успешном завершении транзакции в канале. 0 – прерывание маскировано 1 – прерывание разрешено, это позволяет генерировать прерывание при успешном завершении обмена данными в канале
1	RFLAGNIE		Прерывание при приеме достоверного слова. 0 – прерывание маскировано 1 – прерывание разрешено, это позволяет генерировать прерывание при приеме достоверного ОС в режиме КШ или достоверного КС, ОС или слова данных команды управления в режиме ОУ
0	IDLEIE		Прерывание неактивности контроллера. 0 – прерывание маскировано 1 – прерывание разрешено, это позволяет генерировать прерывание по переходу контроллера в неактивное состояние

20.11.10 Регистр декодирования сообщений MSG

Таблица 157 – Биты регистра декодирования сообщений MSG

Бит	Имя	Значение	Описание
31...14	-		Зарезервировано
13...0	Код сообщения		Регистр содержит код сообщения, полученного в режиме ОУ или М, и доступен только на чтение. В режиме КШ регистр не используется. Регистр обновляется каждый раз при получении нового достоверного КС.

Таблица 158 – Коды сообщений регистра MSG

Код сообщения	CommandWord1	CommandWord2
<i>Команды обмена данными</i>	15:11 10 9:5 4:0	15:11 10 9:5 4:0
0001 Команда приема КШ-ОУ, не групповая	RTA 0 00001-11110 XXXXX	
0080 Команда приема КШ-ОУ, групповая	11111 0 00001-11110 XXXXX	
0004 Команда приема ОУ-ОУ, не групповая	RTA 0 00001-11110 XXXXX	XXXXX 1 00001-11110
0100 Команда приема ОУ-ОУ, групповая	11111 0 00001-11110 XXXXX	не RTA XXXXX 1 00001-11110
0402 Команда передачи ОУ-КШ	RTA 1 00001-11110 XXXXX	
1008 Команда передачи ОУ-ОУ, не групповая	не F XXXXX 0 00001-11110	RTA XXXXX 1 00001-11110
0200 Команда передачи ОУ-ОУ, групповая	11111 0 00001-11110 XXXXX	RTA XXXXX 1 00001-11110
<i>Команда управления</i>		
0410 Код 0-15 K=1 нет данных, не групповая	RTA 1 00000 11111 0XXXX	
0400 Код 0-15 K=1 нет данных, групповая	11111 1 00000 11111 0XXXX	
2420 Код 16-31 K=1 с данными, не групповая	RTA 1 00000 11111 1XXXX	
0040 Код 16-31 K=0 с данными, не групповая	RTA 0 00000 11111 1XXXX	
0800 Код 16-31 K=0 с данными, групповая	1111 0 00000 11111 1XXXX	

20.11.11 Память принимаемых/передаваемых данных DATA

Таблица 159 – Биты регистра DATA

Бит	Имя	Значение	Описание
31...16	-		Зарезервировано
15...0			<p>Данные читаются из памяти или записываются в память в соответствии с подадресом (биты с 9 по 5) достоверного командного слова. Каждому подадресу соответствует 32x16 ячеек памяти на прием и 32x16 ячеек памяти на передачу. Общий объем памяти данных 2Кx16. Физически она состоит из двух банков по 1Кx16.</p> <p>Для подадреса с номером 11110 введено ограничение. При работе в режиме оконечного устройства обращение к данному подадресу при записи со стороны КШ происходит запись не только в память приемника, но и запись в соответствующую ячейку памяти передатчика. Это позволяет организовать режим тестирования ОУ без участия процессора.</p>

## 21 Модуль управления синхронизацией и энергопотреблением

Процессор содержит специальный модуль (CMU) для задания различных режимов работы системы. Возможно индивидуальное задание частоты работы периферийных устройств, частоты процессора, частоты шин обмена. Возможна реализация режимов с пониженным энергопотреблением.

Данный модуль подключен к шине периферийных устройств и имеет базовый адрес **0x800001C0**.

Набор регистров управления/состояния представлен ниже (Таблица 160).

**Таблица 160 – Регистры модуля CMU**

Номер	Обозначение	Разрядность	Тип	Назначение
0	CFG1	32	r/w	Регистр конфигурации периферийных модулей
1	CFG2	16	r/w	Управление синтезатором частот CPU_PLL
2	CFG3	16	r/w	Управление синтезатором частот BUS_PLL
3	CFG4	2	r/w	Управление выбором частот процессора
4	CFG5	32	r/w	Управление синтезатором частот Link_PLL
5	SYS_STS	16	r/w	Флаги состояния / Сброс флагов состояния
7-6	-			
10-8	CFG8	32	r/w	Управление синхронизацией периферийных устройств

Синхронизация процессора осуществляется от внешнего синхросигнала ХТ1. Он используется также для синхронизации часов реального времени RTC.

Процессор имеет несколько частотных доменов, т.е. определенных модулей, функционирующих на собственной частоте:

- процессорное ядро;
- шина периферийных устройств, работающая всегда на половине частоты ядра;
- интерфейс внешней памяти, работающий на собственной частоте;
- часы реального времени, работающие от внешней базовой частоты ХТ1;
- порты связи, работающие на программируемой частоте.

### 21.1 Регистр конфигурации периферийных модулей CFG1

Регистр позволяет задавать различные параметры работы периферийных модулей процессора.

**Таблица 161 – Регистр CFG1**

Биты	Обозначение	По сбросу	Назначение
0	-	0	
1	-	0	
2	LINK_BW	0	Ширина шины линк-портов 0 – ширина шины задается битами регистра управления порта связи. 1 – ширина шины порта связи равна 4 битам

3	SYS_WE	0	Разрешение многократной записи регистров SYSCON,SDRCON 0 – однократная запись 1 – многократная запись
4	ARINC_T0_EN	0	Подключение передатчика 1 контроллера ARINC к внешним контактам: 1 – подключено.
5	ARINC_T1_EN	0	Тоже для передатчика 2
6	ARINC_T2_EN	0	Тоже для передатчика 3
7	ARINC_T3_EN	0	Тоже для передатчика 4
8	MIL_DIS	0	Отключение передатчика МКПД от внешних контактов: 1 – отключен
11-9	-	0	
12	DMA_HP_EN	0	Разрешение высокоприоритетному каналу DMA иметь преимущество перед процессором при доступе к внешней памяти: 1 – разрешено 0 – процессор всегда имеет высший приоритет

## 21.2 Регистры управления внутренними PLL

Процессор имеет синтезатор тактовой частоты CPU\_PLL для формирования синхросигналов процессорного ядра, синтезатор тактовой частоты BUS\_PLL для формирования синхросигнала контроллера внешней памяти, синтезатор тактовой частоты LINK\_PLL для формирования синхросигнала портов связи. Генераторы использует базовую частоту синхронизации XT1.

### 21.2.1 Регистры CFG2, CFG3, CFG5

Регистры доступны по чтению и записи, позволяют задавать различные параметры работы синтезаторов частоты CPU\_PLL, BUS\_PLL, LINK\_PLL.

**Таблица 162 – Регистр CFG2/ CFG3/ CFG5**

Биты	Обозначение	По сбросу	Назначение
3:0	DIVR	0	Вход, задающий коэффициент деления делителя опорной частоты 0000 коэф. деления 1 0001 коэф. деления 2 1111 коэф. деления 16
10:4	DIVF	0	Вход, задающий коэффициент деления делителя обратной связи 0000000 коэф. деления 1 0000001 коэф. деления 3 0000010 коэф. деления 4 1111111 коэф. деления 129
13:11	DIVQ	0	Вход, задающий коэффициент деления выходного делителя 000 коэф. деления 1 001 коэф. деления 2 010 коэф. деления 4 011 коэф. деления 8 100 коэф. деления 16 101 коэф. деления 32

16:14	RANGE	0	Вход подстройки фильтра в зависимости от частоты ЧФД 001 = 10-16 МГц 010 = 16-25 МГц 011 = 25-40 МГц 100 = 40-65 МГц 101 = 65-100 МГц
20	bypass	0	Вход включения режима «bypass» при значении «1» на выход PLLOUT подается сигнал со входа REF.
22:21	S	0	Вход управления выходным мультиплексором, передающим сигнал на выход MUXOUT. 00 – подается сигнал с выхода PLL 01 – подается сигнал с выхода делителя опорной частоты 10 – подается сигнал с выхода делителя в обратной связи 11 – подается сигнал с выхода делителя опорной частоты

Структурная схема PLL приведена ниже (Рисунок 73). Внешний вход частоты ХТІ подключен ко входу REF.

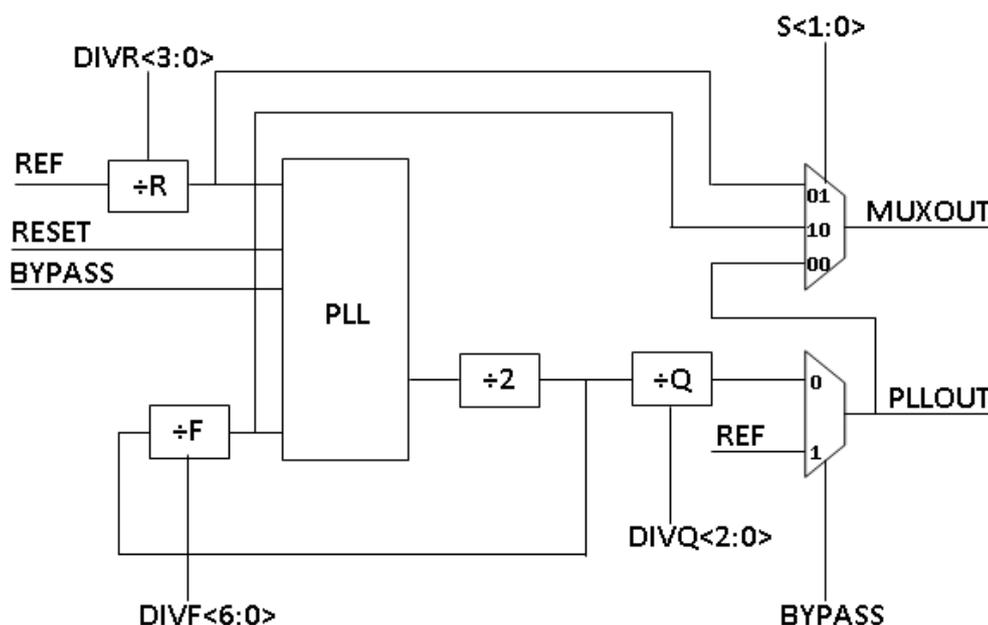


Рисунок 73 – Структура PLL

Процедура работы с CPU\_PLL может быть следующей:

- 1 После сброса системы, синтезатор CPU\_PLL выключен, и процессор работает на входной частоте ХТІ.
- 2 Посредством регистра CFG2 пользователь производит установку рабочих параметров PLL.
- 3 После задержки, необходимой для стабилизации частоты PLL, пользователь устанавливает соответствующий бит в регистре CFG4 и переключается на частоту PLL.
- 4 При необходимости изменить параметры рабочей частоты, пользователь выполняет переключение на частоту ХТІ. Далее выполняются шаги 2 и 3.

Аналогичные действия должны выполняться и при изменениях частоты внешней шины.

### 21.2.2 Регистры CFG4

Регистр позволяют задавать источник частоты различных модулей процессора.

Таблица 163 – Регистр CFG4

Биты	Обозначение	По сбросу	Назначение
0	CPLL_SEL	0	Задание частоты ядра процессора: 0 – XTI 1 – CPLL_out
1	BPLL_SEL	0	Задание частоты интерфейса внешней шины: 0 – XTI 1 – BCLK
2	DIS_CC	0	1 – останов синхронизации ядра
3	DIS_BC	0	1 – останов синхронизации внешнего интерфейса
5:4	BCLK_SEL	00	Выбор источника синхронизации BCLK внешнего интерфейса 00 – частота BPLL_out 01 – частота CPLL_out/2 10 – частота CPLL_out/4 11 – частота CPLL_out/8
6	-	0	
7	LPLL_SEL	0	Задание частоты портов связи 1 – LINK_PLL

Ниже (Рисунок 74) показан процесс формирования частот для процессорного ядра (CCLK), SOC-шины периферийных устройств (SOC-CLK) и для внешней шины (BUS\_CLK). Частота шины периферийных устройств всегда равна половине частоты ядра.

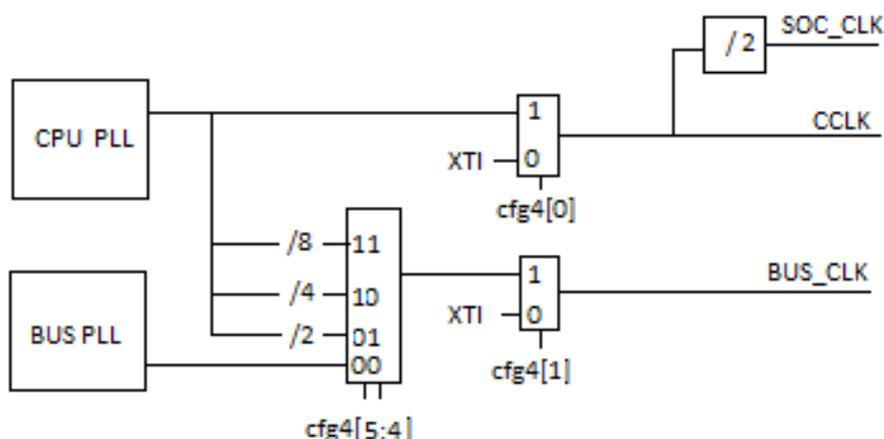


Рисунок 74 – Формирователь частот

Из рисунка видно, что возможны различные варианты задания частоты внешней шины в зависимости от предъявляемых требований. Частота внешней шины может быть оптимально подобрана для получения максимальной частоты обмена с внешними устройствами. После сброса все ресурсы процессора тактируются внешней опорной частотой XTI. Мультиплексы, которые управляются сигналами CFG4[1:0], позволяют изменять источник синхросигнала. Мультиплексор, управляемый сигналами CFG4[5:4], требует чтобы при изменении сигналов CFG4[5:4] выход мультиплексора не был выбран в качестве источника синхронизации BUS\_CLK. Только после задания требуемого значения CFG4[5:4]

можно переключиться на выбранную частоту с помощью сигнала CFG4[1]. Установка бит CFG4[7,1,0] для переключения на частоту PLL модуля оказывает влияние на разрешение записи в конфигурационные регистры. Если, например, ядро процессора переключилось на частоту CPU\_PLL, невозможно изменить значение регистра CFG2. Только после очистки бита CFG4[0] возможна запись в регистр CFG2. Аналогичная защита предусмотрена и для PLL внешней шины и портов связи.

Переключение процессора с одной частоты на другую требует выполнения определенных действий. Подробно процесс переключения излагается в подразделе «Переключение частоты процессора».

Также в процессоре предусмотрена возможность останова частот синхронизации с целью обеспечения реализации режима пониженного энергопотребления. Подробно процедура реализации данного механизма описана в подразделе «Реализация «спящего режима»».

### 21.3 Регистр состояния системы SYS\_STS

Регистр доступен только по чтению. Описание битов регистра состояния SYS\_STS приведено ниже (Таблица 164).

**Таблица 164 – Регистр SYS\_STS**

Бит	Обозначение	Назначение
0	BOOT_0	Состояние входов BOOT[2:0] (они же PC[6:4]) в момент завершения сигнала внешнего сброса или внутреннего сброса по включению питания. Хранят вариант начального старта процессора.
1	BOOT_1	
2	BOOT_2	
3	-	
4	L0_VCMPO	Состояние входа PC[26] в момент завершения сигнала внешнего сброса или внутреннего сброса по включению питания.
5	L1_VCMPO	Состояние входа PC[30] в момент завершения сигнала внешнего сброса или внутреннего сброса по включению питания. Вход используется для задания режима работы контактной площадки внешнего осциллятора.
6	RST	Флаг устанавливается по сигналу внешнего сброса, сброса по включению питания или сброса от сторожевого таймера. Флаг может быть сброшен записью любого значения в регистр SYS_STS.
7	POR	Флаг устанавливается по сигналу внешнего сброса или сброса по включению питания. Флаг может быть сброшен записью любого значения в регистр SYS_STS.
31:8	-	Всегда 0

После сброса регистр состояния позволяет определить был ли сброс вызван внешним источником или включением питания (RST==1, POR==1) или сброс был вызван срабатыванием сторожевого таймера (RST==1, POR==0). В регистре состояние отражается вариант начального старта, который используется внутренним ПЗУ для определения алгоритма загрузки. Если бит L1\_VCMPO равен 1, в качестве опорной частоты используется внешний вход XTI. Если значение бита L1\_VCMPO равно 0, в качестве опорной частоты используется встроенный внутренний осциллятор.

## 21.4 Регистр управления синхронизацией периферийных устройств CFG8

Регистр может использоваться для разрешения или останова синхронизации различных периферийных устройств. Нулевое значение бита разрешает синхронизацию соответствующего периферийного устройства. Единичное значение останавливает синхронизацию.

Таблица 165 – Регистр CFG8

Биты	Обозначение	По сбросу	Разрешение синхронизации для модуля
0	I2S0_en	0	I2S 0 интерфейс
1	I2S1_en	0	I2S 1 интерфейс
2	VCAM_en	0	VCAM
3	SPI_en	0	SPI интерфейс
4	NAND_en	0	NAND контроллер
5	ARINC_en	0	ARINC
6	MIL0_en	0	МКПД 0
7	MIL1_en	0	МКПД 1
8	ADDA0_en	0	UP\DOWN 0
9	ADDA1_en	0	UP\DOWN 1
10	ADDA2_en	0	UP\DOWN 2
11	ADDA3_en	0	UP\DOWN 3
12	GPS0_en	0	Цифровой коррелятор
13	GPS1_en	0	Цифровой коррелятор
14	LCD_en	0	LCD контроллер
15	UART1_en	0	UART1
16	UART2_en	0	UART2

Регистр может использоваться для задания режима пониженного энергопотребления системы. В случае если устройство в данный момент не используется, можно блокировать его частоту синхронизации. После блокировки, доступ к устройству невозможен. После сброса значение регистра равно нулю и на все периферийные устройства поступает сигнал синхронизации.

## 21.5 Переключение частоты процессора

В процессоре предусмотрена возможность гибкого выбора необходимой частоты функционирования. После сброса процессор всегда работает от входного источника синхронизации ХТ1. С помощью имеющихся модулей PLL можно задать требуемую для выполнения задачи частоту процессорного ядра и переключиться на заданную частоту.

Переключение частот выполняется с помощью регистра конфигурации CFG4. Биты регистра номер 7, 1 и 0 выполняют выбор источника синхронизации для портов связи, внешней шины и ядра соответственно. Переключение частот для портов связи и внешнего порта выполняется простой установкой или сбросом соответствующих бит регистра. Главное требование при смене данных частот это обеспечение отсутствия в момент переключения каких-либо действий в передатчиках порта связи или во внешнем интерфейсе.

Перечень необходимых подготовительных действий для смены, например, частоты внешнего интерфейса зависит от конфигурации внешней шины. Если, например, подключена внешняя динамическая память, нужно не забывать о периоде

регенерации динамической памяти. Если смена частоты происходит после сброса процессора и динамическая память еще не инициализирована или нет необходимости обеспечивать сохранность информации в данной памяти, то переключение частоты можно выполнять без подготовительных работ. Если же процессор уже использовал динамическую память и в ней хранится информация, смена частот (например, с высокой на среднюю) будет требовать следующих действий:

- перевод динамической памяти в режим саморегенерации;
- переключение частоты внешней шины на входную частоту XT1;
- подготовка новой частоты в модуле BUS\_PLL;
- корректировка параметров периода регенерации динамической памяти и других временных параметров SDRAM;
- переключение на частоту BUS\_PLL с помощью бита CFG4[1];
- выход SDRAM из режима саморегенерации.

Возможны и другие действия в зависимости от подключенных внешних периферийных устройств и их требований в обслуживании.

Процесс переключения частоты ядра также требует выполнения целого комплекса работ в зависимости от ситуации в которой требуется переключение частот. Рассмотрим самый простой случай когда процессор переключается на заданную частоту сразу после сброса. В этом случае должны быть выполнены следующие действия:

- с помощью регистра CFG2 задается требуемая конфигурация CPU\_PLL для получения необходимой частоты. Выдерживается необходимое время для стабилизации выхода PLL.
- процессор переводится в режим останова синхронизации с помощью установки бита CFG4[2].
- сразу же за установкой бита CFG4[2] выполняется установка бита CFG4[0]. Процессор переключает частоту в момент, когда частоты ядра, периферийных устройств и внешней шины заблокированы.
- через заданный интервал времени частота процессорного ядра разрешается и процессор продолжает выполнение программы с новой частотой.

При переключении частоты ядра используется функция останова синхронизации («спящий режим»). Подробно реализация данного режима и его особенностей приведена в подразделе «Реализация «спящего режима»».

Возможность переключения частоты процессора позволяет гибко выбирать необходимую производительность процессора в зависимости от решаемой задачи и таким образом сокращать потребляемую мощность системы.

## **21.6 Реализация «спящего режима»**

В процессоре возможна реализация режима, в котором частоты ядра, периферийной шины и внешнего интерфейса могут быть остановлены. Данная функция введена для обеспечения режима наименьшего потребления в случае ожидания внешнего или внутреннего события. Ниже показан формирователь частот с учетом функции спящего режима (Рисунок 75). Специальные gate-модули G0, G1, G2 выполняют блокировку выбранных частот.

Установка бита CFG4[3] приведет к останову синхронизации на внешней контактной площадке SCLK и к отсутствию синхронизации внешнего интерфейса.

Данная функция может быть использована, когда выходы внешнего интерфейса выполняют GPIO функции, и нет необходимости выполнять адресуемые обмены по внешней шине. Возобновление синхронизации внешнего интерфейса выполняется программно установкой данного бита в ноль.

Останов синхронизации ядра реализован немного сложнее. При установке бита CFG4[2] в единицу произойдет блокировка прохождения частоты на ядро, шину периферийных устройств и на внешний интерфейс (хотя на внешнем контакте SCLK частота будет присутствовать). После останова синхронизации ядра никакие программы уже невозможно выполнять и выход из данного режима можно выполнить только аппаратно. При этом источники формирующие сигнал пробуждения wake\_up должны использовать другие источники синхронизации.

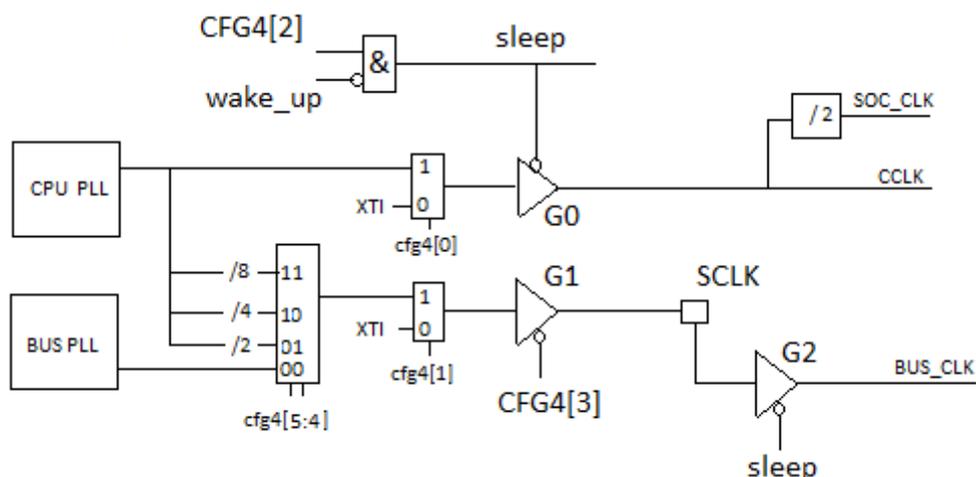


Рисунок 75 – Схема блокировки синхронизации

В качестве событий, которые могут вывести процессор из спящего режима могут использоваться:

- Запрос прерывания по любому из внешних входов портов общего назначения PA, PB, PC. Прерывания должны быть запрограммированы с выбором функции «по уровню».
- Запрос от события «тик-импульс» часов реального времени.
- Запрос от события «прерывание от сторожевого таймера» часов реального времени.
- Запрос от события «будильник» часов реального времени.

Модуль часов реального времени использует для своей работы вход внешней частоты XT1. В связи с этим при останове синхронизации ядра, модуль RTC продолжает работу и позволяет формировать требуемы сигналы.

Переходя в спящий режим, пользователь может запрограммировать выход из состояния сна с помощью необходимого ему события.

События wake\_up, выводящие процессор из состояния сна можно охарактеризовать двумя параметрами: периодом следования и длительностью активного уровня. Если выбран внешний источник порта общего назначения, период следования и длина импульса неизвестны. Если выбран один из источников модуля RTC, параметры данных импульсов известны.

После сброса тик-импульсы формируются с периодом 256 тактов внешней частоты и имеют длительность периода внешней частоты. Период следования можно поменять при помощи значения регистра TIC\_VAL.

Счетчик сторожевого таймера после сброса имеет значение 15, поэтому сторожевой таймер после сброса сформирует первый свой импульс wake\_up через 15 периодов тик-импульса. Дальше период следования импульсов пробуждения от

сторожевого таймера будет равен 256 периодам тик-импульса (если сторожевой таймер не обслуживается).

Будильник часов реального времени будет формировать свой wake\_up запрос каждый раз, когда значение счетчика секунд RTC\_CNT будет совпадать со значением регистра RTC\_MR. Длительность импульса будет равна одной псевдосекунде.

Длительность импульса wake\_up имеет значение, т.к. после пробуждения процессор не сможет опять вернуться в режим сна, если импульс пробуждения не завершился.

Запросы на выход из режима сна от портов общего назначения могут быть заблокированы путем записи в регистр маскирования прерываний значения ноль. Запросы от источников модуля RTC могут быть заблокированы с помощью разрядов 10, 9, 8 регистра управления RTC\_CR.

Из рисунка видно, что процессор может выйти из режима сна, используя запрограммированную частоту, т.е. может возобновить свою работу на частоте ХТІ или частоте CPU\_PLL. Дополнительное выключение модулей PLL для обеспечения пониженного энергопотребления также может быть использовано. Источник внешней синхронизации ХТІ не имеет функции выключения.

Каждый раз, переходя в режим сна, должен обеспечивать гарантию сохранности информации во внешней динамической памяти (если она используется), отсутствие работающих периферийных устройств (их синхронизация будет остановлена), а также должен позаботиться о способе выхода из режима сна.

## 22 Модуль цифрового коррелятора

Модуль цифрового коррелятора (далее ЦК) предназначен для приема информации от внешнего RF-приемника сигналов систем GPS/Glonass, предварительной обработки принятых данных и передачи данных во внутреннюю память процессора. ЦК может принимать сигналы от двух внешних RF-приемников. Внутренняя структура ЦК представляет собой набор из 16-ти идентичных каналов обработки, которые могут работать параллельно.

Внутренний интерфейс модуля ЦК представлен ниже (Таблица 166). Описанные сигналы и шины представляют собой интерфейс внутреннего модуля в системе на кристалле.

Таблица 166 – Интерфейс ЦК

Сигнал	Тип	Функция
PCLK	Вход	Синхросигнал периферийной шины
PRESET	Вход	Сброс (активный 0)
PSEL	Вход	Выбор модуля ЦК (активная 1)
PADDR [11:0]	Вход	Адрес регистра
PWRITE	Вход	Признак записи
PWDATA [31:0]	Вход	Шина данных для записи
PRDATA [31:0]	Выход	Шина данных для чтения
DCLK0	Вход	Синхросигнал RF-приемника 0
SAT0[1:0]	Вход	Данные (sign+magn) приемника 0
DCLK1	Вход	Синхросигнал RF-приемника 1
SAT1[1:0]	Вход	Данные (sign+magn) приемника 1
DMA_WE	Выход	Строб записи данных в контроллер ДМА
DMA_busy	Вход	Признак занятости ДМА
DMA_WA [19:0]	Выход	Адрес для ДМА
DMA_WD [127:0]	Выход	Данные для ДМА
INT	Выход	Запрос прерывания от модуля ЦК

В описанных в таблице сигналах только входы от RF-приемников являются внешними сигналами микросхемы. Регистры модуля доступны программисту посредством интерфейса периферийной шины. Обработанные данные поступают в канал 12 контроллера ДМА, откуда они направляются в запрограммированный каналом буфер. Канал 12 ДМА задает базовый адрес буфера приема. Смещение внутри буфера формируется каналом ЦК и передается в ДМА вместе с данными. Модуль ЦК может формировать запрос прерывания в контроллер прерываний.

### 22.1 Регистры ЦК

Ниже приведен список всех регистров ЦК (Таблица 167). Регистры доступны со стороны шины периферийных устройств и имеют базовый адрес 0x80005000.

**Таблица 167 – Регистры ЦК**

Смещение	Имя	Описание	Сброс
0-7	CH0	Регистры канала 0	
8-15	CH1	Регистры канала 1	
...	...	Регистры канала 2--14	
0x78-0x7F	CH15	Регистры канала 15	
0x80	CR[31:0]	Регистр управления	0
0x81	SR[31:0]	Регистр состояния	0
0x82	LEN[7:0]	Регистр количества (биты 7:0) принимаемых данных. Задаёт количество квадрослов данных которые каждый из каналов выдает в контроллер ДМА до момента переключения в другой режим. Одинаково для всех каналов. Каждое квадрослово содержит 8 пар (два числа по 8 бит) отсчетов.	
0x83	TST[31:0]	Регистр тестовых данных (биты 31:0). Запись в этот регистр вызывает подачу на корреляторы 8-ми пар (2-х бит) данных. Биты 31:16 канал 1, биты 15:0 канал 0	0
0x84	IRQ[15:0]	Регистр запросов прерываний (биты 15:0). Информировать о переключении канала из одного состояния в противоположное. Биты могут быть сброшены записью 1-го значения.	
0x85	CFG[3:0]	Регистр конфигурации модуля ЦК.	

Каждый из каналов может работать с одним из двух буферов. После выгрузки заданного количества квадрослов входных отсчетов (задается регистром LEN) канал переключается на работу с другим буфером. При этом он использует другой набор регистров.

Соответственно, при работе с другим буфером канал может выбрать другой источник сигнала, значение задержки, а также значения шагов.

Длина выгружаемых данных для всех каналов одинакова.

При переключении канала вырабатывается флаг прерывания. Канал может быть обслужен и ему можно задать новые параметры.

### 22.1.1 Регистр управления CR

Регистр позволяет задать основные режимы работы модуля. Назначение бит регистра приведено ниже (Таблица 168).

**Таблица 168 – Регистр управления**

Бит	Название	Функция
0	EN0	Разрешение работы канала 0 (1)
...		
15	EN15	Разрешение работы канала 15 (1)
16	IE0	Разрешение прерывания от канала 0 (1 – разрешено)
...		
31	IE15	Разрешение прерывания от канала 15 (1 – разрешено)

После сброса значение регистра управления равно 0.

### 22.1.2 Регистр состояния SR

Позволяет анализировать состояние модуля. Назначение бит регистра приведено ниже (Таблица 169).

**Таблица 169 – Регистр состояния**

Бит	Название	Функция
0	PPS0	1 – канал 0 работает с буфером 1 0 – канал 0 работает с буфером 0
...	...	..
15	PPS15	1 – канал 15 работает с буфером 1 0 – канал 15 работает с буфером 0
16	Err0	1 – ошибка в работе канала 0 (переполнение) 0 – нет ошибок
..	...	..
31	Err15	1 – ошибка в работе канала 15(переполнение) 0 – нет ошибок

### 22.1.3 Регистр количества LEN

Позволяет задавать количество отгружаемых каналом квадрослов данных до момента переключения в другой режим. Назначение бит регистра приведено ниже (Таблица 170).

**Таблица 170 – Регистр LEN**

Бит	Название	Функция
7:0	LEN	Количество квадрослов данных минус 1
31:8	-	Не используются. При чтении всегда ноль

### 22.1.4 Регистр конфигурации CFG

Позволяет задавать режим работы модуля ЦК. Назначение бит регистра приведено ниже (Таблица 171).

**Таблица 171 – Регистр TSM**

Бит	Название	Функция
0	TESTM	Разрешение тестового режима: 1 – разрешено. 0 – запрещено. Нормальный режим работы
1	NPS0	Выбор активного фронта синхросигнала приемника 0: 0 – переход из 0 в 1 1 – переход их 1 в 0
2	NPS1	Выбор активного фронта синхросигнала приемника 1: 0 – переход из 0 в 1 1 – переход их 1 в 0
3	-	Бит общего назначения
31:4	-	Не используются. При чтении всегда ноль

### **22.1.5 Регистр тестовых данных TST**

Позволяет задавать данные для работы модуля ЦК без использования внешних приемников. Регистр имеет 32 разряда и младшие 16 разрядов используются как данные от приемника 0, а старшие как данные от приемника 1. Разрядность данных приемника равна 2-м битам. В модуле ЦК эти данные поступают на линию задержки имеющей восемь стадий. Использование 16 разрядов данных тестового регистра позволяет задать состояние всей линии задержки конкретного приемника. Каждый канал обрабатывает 2-х битные входные данные с конкретной стадии линии задержки. Номер стадии программируется.

### **22.1.6 Регистр запросов прерываний IRQ.**

Отражает состояние запросов прерываний от каждого из каналов ЦК. Модуль ЦК имеет один общий запрос прерывания в контроллере прерываний, поэтому для определения номера канала вызвавшего запрос прерывания может использоваться регистр IRQ. Биты с 0 по 15 регистра соответствуют каналам с 0 по 15 модуля ЦК. Старшие 16 бит не используются и при чтении равны 0. Регистр запросов доступен по чтению. При записи возможен только сброс бит регистра. Если при записи какой-то из младших 16-ти разрядов данных равен 1, соответствующий ему разряд регистра IRQ будет сброшен в ноль. Нулевое значение разряда данных не влияет на бит регистра.

## **22.2 Регистры канала**

**Таблица 172 – Регистры канала**

<b>Смещение</b>	<b>Имя</b>	<b>Описание</b>	<b>Сброс</b>
0	SC0_CNT	27 бит регистр-счетчик таблицы синуса-косинуса	
1	SC0_STEP	32 бит регистр приращения счетчика SC0_CNT	
2	DZ0_CNT	27 бит регистр-счетчик дециматора	
3	DZ0_STEP	32 бит регистр приращения счетчика DZ0_CNT	
4	SC1_CNT	27 бит регистр-счетчик таблицы синуса-косинуса	
5	SC1_STEP	27 бит регистр приращения счетчика SC1_CNT	
6	DZ1_CNT	27 бит регистр-счетчик дециматора	
7	DZ1_STEP	27 бит регистр приращения счетчика DZ1_CNT	

### **22.2.1 Регистры SCx\_CNT и SCx\_STEP.**

Регистр SC\_CNT содержит значение указателя на таблицу синусов-косинусов. Начальное значение указателя можно задать путем записи в регистр. Каждый раз после чтения константы из таблицы (обработка одного принятого значения), значение указателя наращивается на значение в регистре SC\_STEP, т.е. выполняется операция  $SC\_CNT = SC\_CNT + SC\_STEP[26:0]$ . Таблица значений синусов-косинусов имеет восемь элементов. Для выбора одного из 8-ми значений используются только биты 26:24 регистра SC\_CNT. Значения синусов и косинусов приведены ниже (Таблица 173).

Таблица 173 – Значение таблицы синусов косинусов.

SC_CNT [26:24]	SIN (SATi)				COS(SATi)			
	00	10	01	11	00	10	01	11
0	1	-1	3	-3	2	-2	6	-6
1	2	-2	6	-6	1	-1	3	-3
2	2	-2	6	-6	-1	1	-3	3
3	1	-1	3	-3	-2	2	-6	6
4	-1	1	-3	3	-2	2	-6	6
5	-2	2	-6	6	-1	1	-3	3
6	-2	2	-6	6	1	-2	3	-3
7	-1	1	-3	3	2	-1	6	-6

Поскольку только 27 младших бит используются для значения приращения, оставшиеся биты используются для задания других функций канала. Бит SC\_STEP[31] выбирает источник сигнала (0 – приемник 0, 1 – приемник 1). Биты SC\_STEP[30:28] представляют собой 3-х битовое значение, которое используется для выбора стадии линии задержки, с которой выполняется прием данных для обработки. Пара регистров 0 используется для работы в текущем режиме, а пара 1 для работы в следующем режиме.

### 22.2.2 Регистры DZ\_CNT и DZ\_STEP

Данные регистры используются для задания коэффициента децимации. Каждое входное значение с приемника сигнала, после обработки с использованием значений синуса и косинуса, поступает в аккумулятор сигналов, который выполняет накопление поступающих значений. Значение регистра DZ\_CNT позволяет задать количество значений, которые поступают на аккумулятор, для формирования итоговой суммы. Счетчик после каждого такта накопления увеличивает свое значение на приращение DZ\_STEP, т.е. выполняется операция  $DZ\_CNT = DZ\_CNT + DZ\_STEP$ . Момент переполнения счетчика DZ\_STEP, т.е. загрузка в DZ\_CNT значения которое имеет старший бит (26-й) равным нулю в момент, когда значение бита DZ\_CNT[26] равно 1, вызывает окончание текущего накопления, формирование 2-х отсчетов и начало новых накоплений. Накопленная пара отсчетов поступает в выходной буфер. Когда будут сформированы 8-мь пар отсчетов (128-бит квадрослово), канал выполнит запрос к контроллеру ДМА и передаст подготовленное значение. При этом канал также подготовит значение смещение адреса, по которому должна быть выполнена запись данных. Формирование смещения происходит следующим образом. После старта канала его внутренний счетчик QW\_CNT начинает отсчет переданных квадрослов. Значение счетчика изменяется от нуля до значения заданного в регистре LEN, т.е. максимальный диапазон значений от 0 до 255. Состав смещения показан ниже (Таблица 174).

Таблица 174 – Структура смещения канала

Биты	Значение	Описание
1:0	00	Адрес слова в квадрослове
9:2	QW_CNT	Номер квадрослова
13:10	CH_N	Номер канала
14	PPS	Выбор одного из 2-х буферов. Значение, с которым работает буфер, задается в регистре состояния.
19:15	00000	Всегда нули.

Таким образом, зная номер канала и номер буфера с которым он работает, а также зная максимальное количество формируемых отсчетов, можно определить местонахождение данных в оперативной памяти и выполнить дальнейшую обработку.

После выгрузки заданного значения отсчетов канал переключается на работу со вторым буфером, и первый становится доступным для обработки. При этом для второго буфера канал использует новый набор регистров управления.

Факт окончания выгрузки данных в текущий буфер отражается в регистре запросов прерывания и может использоваться для выполнения процессором действий по обслуживанию канала.

Все сформированные данные передаются во внутреннюю оперативную память через ДМА канала номер 12. В канале 12 задается базовый адрес массива данных модуля ЦК. Скорость работы канала должна быть достаточной для обслуживания всех 16 каналов ЦК. Если в процессе обслуживания какой-то канал не сможет передать свои данные, будет сформирован признак ошибки.

## 23 Порт JTAG и интерфейс отладки

Процессор поддерживает IEEE Standard 1149.1 Joint Test Action Group (JTAG) порт для тестирования системы. Этот стандарт определяет метод последовательного сканирования состояния входов-выходов для каждого компонента системы. Последовательный порт JTAG также используется эмулятором для получения доступа к встроенным возможностям отладки процессора.

Описываемые ниже функции реализованы в аппаратной части процессора для того, чтобы позволить программе-отладчику и операционным системам выполнять более сложные задачи. Функциональность разбита на группы по нескольким уровням. Эти уровни выстроены друг над другом, что позволяет отладчику или ядру ОС осуществлять более качественный контроль и анализ процессора.

Отладчик может осуществлять контроль процессора двумя путями:

- используя отладочный монитор (работающий в режиме супервизора);
- используя встроенный эмулятор (ICE).

При отладке с помощью программного монитора в режиме супервизора, некоторые системные ресурсы используются только монитором (например, память, канал коммуникации с хостом, прерывание, флаги и т. п.).

При отладке с использованием встроенного эмулятора, используется резервный коммуникационный канал (порт доступа к тестированию JTAG) для контроля за процессором.

Порт JTAG является подчиненным устройством на SOC-шине. Многие отладочные функции являются востребованными как при использовании программного метода отладки, так и при аппаратном.

Функции и возможности процессора для поддержки отладки программ включают:

- **Точки наблюдения**

Позволяет пользователю указывать конкретный диапазон адресов точки наблюдения и условия, которые останавливают процессор. После остановки, пользователь может проанализировать состояние процессора, восстановить прежнее состояние и продолжить выполнение команд. Программные точки останова и одношаговые операции также выполняются с помощью точки наблюдения.

- **Запись истории**

Специальный буфер трасс позволяет проследить за программным счетчиком с целью восстановления хода программы. Буфер трассировки на кристалле сохраняет последние 32 перехода (все дискретные значения счетчика программ) выполненные устройством управления ядра, позволяя восстановить последний программный путь.

- **Счетчик тактов**

Обеспечивает функцию подсчета количества процессорных тактов для всех функций программного кода.

- **Мониторинг производительности**

Позволяет осуществлять мониторинг некоторых внутренних событий ядра процессора при выполнении программного кода. Отладчик может точно указать, какое из внутренних событий нуждается в мониторинге.

- **Доступ к защищенным регистрам**  
Защищенные регистры доступны только в режиме супервизора или ICE. К ним нет доступа в режиме пользователя.
- **Счетчики точки наблюдения**  
Позволяет пользователю искать *n*-е возникновение события до остановки программы.
- **Исключение**  
Событие, которое отменяет выполнение всех последующих команд.

## 23.1 Рабочие режимы

Процессор работает в одном из трех режимов: режим эмулятора, супервизора или пользователя. В режимах пользователя и супервизора все команды выполняются стандартно. Тем не менее, в режиме пользователя ограничен доступ к регистрам. В режиме эмуляции имеются ограничения на использование некоторых типов команд.

## 23.2 Ресурсы отладки

В процессоре предусмотрены несколько ресурсов отладки, включая:

- специальные команды;
- точки наблюдения;
- буфер трассировки адреса команды.

### 23.2.1 Специальные команды

Процессор поддерживает специальные команды, которые используются для помощи в отладке системы. Данные команды нужны для выполнения программных остановов в отладчиках и вызовов операционной системы.

### 23.2.2 Точки наблюдения

Точки наблюдения адреса позволяют пользователю проверять состояние процессора во всех случаях, когда происходит выполнение условий доступа к памяти, определенной пользователем. Существует три точки наблюдения, которые работают параллельно. Во время работы каждой точки наблюдения пользователь может определять условия срабатывания при доступе к шине и действия, которые должен выполнить процессор.

Точка наблюдения имеет два адресных регистра: WPiL и WPiH. Когда точка наблюдения запрограммирована на один адрес, используется только регистр с младшим адресом (WPiL). Если точка наблюдения запрограммирована на диапазон адресов, в WPiL содержится младший адрес, а в WPiH – старший адрес. Необходимо установить регистры указателя адреса до того, как установлен регистр управления WPiCTL, который переводит точку наблюдения в активное состояние. Адресные регистры точки наблюдения не могут быть изменены, пока точка наблюдения активна.

Работа точки наблюдения определяется ее регистром управления WPiCTL, (где: *i* – это 0, 1 или 2 в зависимости от номера точки наблюдения).

Следующая информация должна быть запрограммирована в регистрах до начала поиска:

- адрес или диапазон адресов;
- отсчет.

Когда регистр управления устанавливается в активное состояние, поле отсчета устанавливается в начало отсчета. С этого момента аппаратная часть точки наблюдения отслеживается внутренние шины с целью совпадения адресов передач, что указывается в регистрах точек наблюдения.

При каждом совпадении значение счетчика уменьшается. После того, как счетчик достигает значения нуля, инициируется исключение: программное исключение или вызов эмулятора. Выбор действий определяется разрядами 3-2 регистра WPiCTL.

С этого момента регистр состояния указывает на завершение работы точки наблюдения. Чтобы начать новый поиск, пользователю необходимо снова записать регистр управления. Три точки наблюдения связаны с тремя различными шинами.

Точка наблюдения 0 используется для мониторинга на I-шине. Точка наблюдения 0 поддерживает пошаговую опцию (SSTP), которая используется при эмуляции для выполнения одношаговых операций.

Точка наблюдения 1 используется для наблюдения за доступом к данным через J- и K-шины. При этом мониторинг может выполняться для одной из шин или для обеих сразу. Точка наблюдения 1 поддерживает также выбор типа доступа: только чтение, только запись, чтение и запись.

Точка наблюдения 2 используется для наблюдения за доступом к данным через S-шину. Точка наблюдения 2 поддерживает также выбор типа доступа: только чтение, только запись, чтение и запись.

Регистр WPiSTAT указывает на текущее состояние точки наблюдения. Он хранит текущее значение счетчика поиска и режима работы точки наблюдения. Описание разрядов регистра состояния приведено ниже (Таблица 175).

**Таблица 175 – Описание разрядов регистра WPiSTAT**

Разряды	Имя	Описание
15–0	VALUE	Значение точки наблюдения
		Определяет текущее значения счетчика точки наблюдения
17–16	EX	Разряд выполнения
		Выполнение точки наблюдения 00 – точка наблюдения запрещена 01 – поиск совпадений 11 – окончание работы счетчика точки наблюдения
31–18	Reserved	Резервные

### **23.2.3 Буфер трассировки адреса команды (TBUF)**

Поскольку программный счетчик недоступен вне кристалла в реальном времени в процессе выполнения команд, то буфер трассировки используется для помощи в отладке программы.

Буфер трассировки включает в себя 32 регистра, в которых хранится история последних от 16 до 32-х переходов, выполненных устройством управления ядра процессора, что дает возможность пользователю полностью восстановить путь выполнения программы в течение последних переходов.

Переходы циклично записываются в регистры буфера трассировки. Первый записывается в TRCB0, следующий в TRCB1, и т. д. до TRCB31. Тридцать второй переход снова записывается в TRCB0, и так далее. Для того, чтобы определить какой из регистров был записан последним, пользователь должен обратиться к регистру указателя буфера трассировки, который указывает на последний записанный вход.

Буфер трассировки всегда запоминает адрес команды вызвавшей переход. Если адрес перехода вычислялся с использованием дополнительного регистра (не только РС и смещение в коде команды), то кроме адреса команды перехода в буфере трасс запоминается и адрес перехода.

Для того, чтобы различать какой тип информации записан в линии буфера трасс, используется регистр маски TRCBMASK. Каждый разряд в RCBMASK связан с соответствующим регистром TRCBi. Если он сброшен, TRCBi хранит адрес команды перехода. Если же он установлен, TRCBi содержит адрес перехода.

### **23.3 Мониторинг производительности**

Целью мониторинга производительности является оптимизация функционирования рабочих приложений. Аппаратные возможности процессора обеспечивают подсчет числа циклов, которые были затрачены на выполнение кода программы или подсчет числа заданных событий при выполнении программы. Мониторинг обеспечивается с помощью трех регистров: счетчик циклов, маска монитора производительности и счетчик монитора производительности.

Счетчики циклов и монитора производительности должны быть установлены пользователем в ноль перед тем, как начнется определенный период работы. Соотношение между общим счетчиком и счетчиком монитора производительности и дает требуемое значение.

Когда счетчик монитора производительности переполняется, это вызывает исключительную ситуацию.

#### **23.3.1 Регистр маски монитора производительности (PRFM)**

Регистр маски монитора производительности (PRFM) определяет, какие события в процессоре отслеживаются и подсчитываются счетчиком монитора производительности (PRFCNT). Значением сброса регистра PRFM является 0x0000 0000. Подробное описание разрядов регистра приведено ниже (Таблица 176).

**Таблица 176 – Биты регистра PRFM**

<b>Бит</b>	<b>Имя</b>	<b>Назначение</b>
7:0	-	Резерв
8	Jexe	Подсчет команд выполненных на линии конвейера J 1 – включен 0 – выключен
9	Kexe	Подсчет команд выполненных на линии конвейера K 1 – включен 0 – выключен
10	Xexe	Подсчет команд выполненных на линиях конвейера X1 и X2 1 – включен 0 – выключен
11	Yexe	Подсчет команд выполненных на линиях конвейера Y1 и Y2 1 – включен 0 – выключен

12	Sexe	Подсчет команд выполненных на линии конвейера S 1 – включен 0 – выключен
15:13	-	резерв
16	STALL	Подсчет команд остановов конвейера вызванных ожиданием данных или другими блокирующими конвейер ситуациями 1 – включен 0 – выключен
17	BTB_true	Подсчет количества правильных предсказаний буфера переходов 1 – включен 0 – выключен
18	ABORT	Подсчет количества программных исключительных ситуаций 1 – включен 0 – выключен
19	Uexe	Подсчет количества линий команд выполненных в режиме пользователя 1 – включен 0 – выключен
20	BTB_false	Подсчет количества ошибочных предсказаний буфера переходов 1 – включен 0 - выключен
21	BTB_load	Подсчет количества загрузок в буфер переходов 1 – включен 0 – выключен
31:22	-	Резерв

Все события, которые отслеживаются монитором производительности суммируются по «ИЛИ» и при их наступлении происходит увеличение счетчика PRFM на 1. Логичным является разрешение подсчета одного условия из многих.

### **23.3.2 Регистр счетчика монитора производительности (PRFCNT)**

Назначение счетчика производительности – увеличивать свое значение на 1 каждый раз, когда происходит отслеживаемое событие. Отслеживаемое событие определяется регистром маски монитора производительности (PRFM). Счет прекращается, когда процессор переходит в режим эмуляции. Регистр очищается после сброса.

### **23.3.3 Регистры счетчика циклов (CCNTx)**

Длина счетчика циклов равна 64 разряда, но доступен он как два универсальных 32-разрядных регистра – CCNT0 и CCNT1. После сброса значение счетчика равно нулю.

В связи с тем, что счетчик 64 бита, а доступ возможен только к частям регистра, необходимо соблюдение специальной процедуры чтения и записи регистра. При чтении сначала считывается нижняя часть (CCNT0), а затем верхняя часть (CCNT1). При чтении нижней части верхняя копируется в буфер и это гарантирует ее корректное значение. При записи сначала пишется нижняя часть (она попадает в буфер), а затем верхняя. При записи верхней полное 64-битное значение попадает в счетчик.

Счет прекращается, когда процессор переходит в режим эмуляции.

### **23.3.4 Регистр данных точки наблюдения 1 (WPDR)**

Точка наблюдения 1 имеет дополнительные возможности по мониторингу обмена данными ядра процессора с памятью. Выше описывался механизм срабатывания исключительной ситуации в случае совпадения адресов точки наблюдения с адресами шин J и K. Однако имеется возможность дополнить сравнение адресов еще и сравнением данных. При чтении имеется возможность контролировать то, какие данные считывает процессор. Если дополнительно к совпадению адресов происходит совпадение и прочитанных данных – точка срабатывает. Также можно контролировать и процесс записи данных. Необходимые для контроля данные записываются в регистр WPDR. Необходимо отметить, что шина чтения или записи имеет разрядность 128 бит. Разрядность регистра данных только 32 бита. Поэтому сравнение выполняется только для 32-разрядного слова шины данных определяемого младшими битами шины адреса, независимо от того выполняется чтение или запись 64-х или 128-разрядного слова.

### **23.3.5 Регистр маски точки наблюдения 1 (WPMR)**

При использовании шины данных в описании точки наблюдения 1, не все биты шины данных могут понадобиться при анализе читаемых или записываемых данных. Регистр маски позволяет установить, какие биты регистра данных WPDR должны участвовать в сравнении. Если бит установлен в 1, соответствующий ему бит данных сравнивается с битом шины данных. После сброса регистр маски равен нулю и данные не участвуют в работе точки наблюдения.

## **23.4 JTAG интерфейс**

Процессор поддерживает порт доступа к средствам тестирования, соответствующий стандарту IEEE 1149.1.

### **23.4.1 Выводы JTAG порта**

Таблица 177 описывает выводы процессора Процессор, используемые в аппаратной части эмуляции JTAG. Подчеркнутые снизу названия выводов и выводы с чертой над названием указывают на активные низкие сигналы.

**Таблица 177 – Выводы порта связи и эмуляции устройства ввода/вывода**

<b>Сигнал</b>	<b>Тип</b>	<b>Описание</b>
TRST	вход	Сброс JTAG Сбрасывает интерфейс JTAG. Сигнал TRST должен быть активирован после включения питания, чтобы убедиться в правильной работе JTAG. Вход TRST имеет внутренний повышающий резистор
TCK	вход	Синхронизация JTAG Обеспечивает тактовый сигнал для работы JTAG
TDI	вход	Входной сигнал данных JTAG Имеет внутренний повышающий резистор
TDO	выход	Выходной сигнал данных JTAG
TMS	вход	Выбор режима тестирования JTAG Управляет машиной состояний тестирования. Имеет внутренний повышающий резистор
EMU	выход	Эмуляция Признак перехода процессора в режим эмуляции

В дополнение к стандартным функциям JTAG, выводы TMS и EMU выполняют следующие функции отладки:

- Вход TMS (когда разряд TЕМЕ установлен в регистре EMUCTL) выполняет функцию запроса к процессору на переход в режим эмуляции. Исключение эмуляции формируется при каждом нарастающем фронте сигнала TMS. Данная функция является дополнительной к функциональности TMS в JTAG.
- Выходной сигнал EMU является сигналом с открытым стоком, который активизируется, только если разряд EMUOE установлен в регистре EMUCTL. Вывод EMU указывает на следующее:
  - Произошло срабатывание точки наблюдения, но регистр управления имеет в поле типа (поле EXTYPE в WPiCTL) значение ноль. Уровень сигнала EMU в этом случае снижается на 20 циклов CCLK.
  - Каждый раз, когда процессор находится в режиме эмуляции и готов к выполнению новых линии команд, подаваемой в регистр EMUIR, уровень сигнала на выводе EMU снижается до тех пор, пока такая команда добавляется.

### 23.4.2 Группа регистров эмулятора JTAG

**Таблица 178 – Группа регистров JTAG**

<b>Имя</b>	<b>Описание</b>	<b>Адрес</b>	<b>Значение по умолчанию</b>
EMUCTL	Управление	0x 03A0	0x0000 0000
EMUSTAT	Состояние	0x 03A1	0x0000 0002
EMUDAT	Данные	0x 03A2	Не определено
IDCODE	Код ID	0x 03A3	0x427A F0CB
EMUIR:0	Команда 0	0x 03A4	0x33C0 0000
EMUIR:1	Команда 1	0x 03A5	0x33C0 0000
EMUIR:2	Команда 2	0x 03A6	0x33C0 0000
EMUIR:3	Команда 3	0x 03A7	0xB3C0 0000
-	-	0x 03A8	-
OSPID	OSPID	0x 03A9	Не определено
-	-	0x 03AA– 0x 03BF	-

#### 23.4.2.1 Регистр управления (EMUCTL)

Регистр EMUCTL устанавливает параметры эмуляции. Значением сброса регистра EMUCTL является 0. Большинство разрядов регистра EMUCTL используются эмулятором и не должны модифицироваться программой. Единственными разрядами, предназначенными для использования программой, являются SWRST (программный сброс) и BOOTDIS (запрет загрузки). Подробное описание разрядов регистра приведено ниже (Таблица 179).

Таблица 179 – Регистр EMUCTL

Бит	Имя	Назначение
0	EMEN	Включение эмулятора 1 – включен 0 – выключен
1	TEME	Разрешение генерации запроса к процессору по входу TMS: 1 – разрешено 0 – запрещено
2	EMUOE	Разрешение активной выдачи информации на выход TDO 1 – разрешено 0 – запрещено
3	-	Бит общего назначения
4	SWRST	Запрос программного сброса: 1 – сброс 0 – рабочий режим
5	BOOTDSBL	Запрещение загрузки 1 – запрещено 0 – разрешено Установка данного бита запрещает загрузку из внешнего EPROM.
31:6	-	Всегда ноль

#### 23.4.2.2 Регистр состояния эмуляции (EMUSTAT)

Регистр EMUSTAT отражает состояние эмуляции. Значением сброса регистра EMUCTL является 0x0000 0002. Подробное описание разрядов регистра приведено ниже (Таблица 180).

Таблица 180 – Регистр EMUSTAT

Бит	Имя	Назначение
0	EMUMOD	1 – процессор в состоянии эмуляции 0 – нормальный режим работы
1	IRFREE	Флаг готовности команды в регистре EMUIR 1 – команда готова 0 – нет
2	INRESET	Флаг сброса эмулятора
31:3	-	Всегда ноль

#### 23.4.2.3 Регистр типа кристалла и кода ID (IDCODE)

Регистр IDCODE определяет ID модели процессора и типа кристалла. Регистр доступен только по чтению и имеет значение 0x427AF0CB.

#### 23.4.3 Регистр команды JTAG

Регистр команды JTAG имеет длину, равную пяти разрядам, и позволяет контроллеру доступа порта (TAP) выбрать любой из сканируемых регистров данных. Младший значащий бит первым выдвигается на TDO. Кодирование команд приведено ниже (Таблица 181).

Таблица 181 – Декодирование команды доступа порта JTAG

Если значение IR4=0 равно...	Соответствующая команда...	выполнить...
00000	EXTEST	выполнить EXTEST
10000	Sample/Preload	Выполнить

		сэмплирование и предзагрузку
01000	EMUIR	Сканирование в регистре EMUIR
10100	EMUDAT	Сканирование в регистре EMUDAT
01110	IDCODE	Сканирование кода регистра ID
11110	SAMPLEPC	Сэмплирование ПК, к которому подключен процессор
10001	EMUTRAP	Выполнить перехват эмуляции
10011	OSPID	Сканирование регистра OSPID
00100	EMUCTL	Сканирование регистра EMUCTL
01100	EMUSTAT	Сканирование регистра EMUSTAT
11111	BYPASS	Параллельный режим

#### **23.4.4 Регистры данных**

Регистры данных выбираются через регистр команд. Как только соответствующее значение регистра данных записывается в регистр команд, и состояние TAP изменяется на SHIFT-DR, данные выбранного регистра начинают выдвигаться на TDO и приниматься с TDI. Больше деталей в спецификации стандарта IEEE 1149.1.

Регистры данных:

– **Обход (Байпас)**

Регистр, отвечающий требованиям IEEE 1149.1, является одноразрядным регистром.

– **Пограничные регистры**

Регистры, используемые многими командами JTAG. Все три команды JTAG соответствуют требованиям IEEE 1149.1

– **EMUIR**

Команда JTAG загружает 48 разрядов, из которых 32 наименьших значащих разряда загружаются в регистр EMUIR. EMUIR является 128-разрядным регистром. В режиме эмуляции регистр загружается четыре раза последовательно (одной командой JTAG) или загружается до тех пор, пока в загружаемом слове не установится разряд EX (тогда остальные слоты EMUIR считаются командами NOP). Каждый раз, когда данные загружаются в EMUIR, сначала загружаются разряды 31–0, затем загружаются разряды 63–32 и так далее.

Когда четыре загрузки EMUIR завершены (или линия команд была завершена установленным разрядом EX) команды, загруженные в JTAG, передаются в устройство управления ядра и выполняются.

Когда счетверенное слово загружается в EMUIR, оно должно включать одну полную линию команд. Линия команд не должна продолжаться в

следующем счетверенном слове. Лишние слоты должны быть заполнены NOP командами.

Запись в EMUIR является недопустимой операцией, когда процессор не находится в режиме эмуляции.

– **EMUDAT**

Это 48-разрядный регистр, в котором 32 старших значащих разряда используются для сканирования Ureg с таким же именем. Эмулятор использует этот регистр для чтения и записи всех остальных регистров в процессоре.

- Эмулятор также использует данный регистр для доступа к памяти. Должна быть команда, использующая IALU для генерации адреса доступа к памяти, назначением или источником которой является EMUDAT. Когда эмулятор выполняет серию доступов к памяти, процессор должен вставить циклы удержания для того, чтобы завершить все доступы (как во время нормальной работы).

## 24 Начальный старт процессора

Разряды порта PC[6:4] (они же BOOT[2:0]) во время сброса процессора задают режим начального старта процессора. Процессор может выполнить один из 8-ми вариантов начального старта. В любом из выбранных вариантов, процессор начинает работу с выполнения кода программы размещенного во внутреннем загрузочном ПЗУ. Код программы начального старта приведен ниже.

```
boot_start:
    // Enable multiple writes to SYSCON/SDRCON
    j0 = CFG_APB_SYS_WE;;
    [j31 + CMU_CFG1_LOC] = j0;;

    nop;;

    // Define boot mode
    j1 = [j31 + CMU_SYS_STS_LOC];;
    j1 = j1 and SYS_STS_BOOT0 | SYS_STS_BOOT1 | SYS_STS_BOOT2;;

    comp(j1, 7);;
    if jeq, jump start_LINK8(np);;
    comp(j1, 6);;
    if jeq, jump start_LINK4(np);;
    comp(j1, 5);;
    if jeq, jump start_LINK1(np);;
    comp(j1, 4);;
    if jeq, jump start_NAND(np);;
    comp(j1, 3);;
    if jeq, jump start_SPIF(np);;
    comp(j1, 2);;
    if jeq, jump start_XPF(np);;
    comp(j1, 1);;
    if jeq, jump start_IRQ(np);           j10 = 0x00;; // 32-bit mode
    j10 = SYSCON_MEM_WID64;;           // 16-bit mode
    jump start_IRQ(np);
```

Как следует из начального текста программы, процессор читает код старта из регистра состояния и выполняет переход на одну из меток, соответствующей номеру начального старта. Далее будут подробно рассмотрены действия которые выполняются при выборе того или иного варианта старта.

### 24.1 Загрузка из внешней NAND флэш памяти

Если выбран вариант старта номер 4, процессор выполняет переход на метку start\_NAND, с которой начинается выполнение следующего кода программы:

```
start_NAND:

    // Disable clocks that are not required
    j6 = ~CMU_CLKDIS_NAND;;
```

```

[j31 + CMU_CLKDIS_LOC] = j6;;

// Setup GPIO
// Alternate function for DATA and ADDR bus
j0 = PX_ALT_PDB2 | PX_ALT_PDB3;; // PXD[31:16] = alternate functions
j0 = j0 or PX_ALT_PDB23F0;;      // PXD[31:16] = NAND FLASH
[j31 + PX_ALT_LOC] = j0;;

// NAND controller is active after reset. It's settings are:
//   - page size = 2K
//   - 3 bytes to select a page (row)
//   - 2 bytes to select a column
//   - 5 address bytes and 2 read commands

// Setup number of 32-bit words to read
lc0 = 256;;
[j31 + NAND_CNTR_LOC] = lc0;;
j4 = 0x00;;      // start address

read_NAND:
j0 = [j31 + NAND_SR_LOC];;
j0 = j0 and NAND_SR_EMPTY;;
if njeq, jump read_NAND;;
xr0 = [j31 + NAND_DR_LOC];;
[j4 += 0x01] = xr0;;
if nlc0e, jump read_NAND;;

// Jump to target address
cjmp = 0x00;;
cjmp(abs)(np);;

```

Код программы выполняет загрузку 256 слов данных из внешней NAND флэш-памяти во внутреннюю память, начиная с адреса 0. После завершения загрузки управление передается на адрес 0.

## **24.2 Загрузка EPROM**

Если выбран вариант старта номер 2, процессор выполняет переход на метку start\_XPF, с которой начинается выполнение следующего кода программы:

```

start_XPF:

// Disable clocks that are not required
j6 = 0xFFFFFFFF;;
[j31 + CMU_CLKDIS_LOC] = j6;;

// Setup GPIO
// Alternate function for nBMS and nRD signals
j0 = (PF_ALT << 20) | (PF_ALT << 21);;
// Alternate function for MS1 and MS0 signals
j0 = j0 or (PF_ALT << 18) | (PF_ALT << 19);;
[j31 + GPC_ALT_LOC] = j0;;

```

```

// Alternate function for DATA and ADDR bus
j0 = PX_ALT_PDB0;; // data (8-
bit)

j0 = j0 or PX_ALT_PAB0 | PX_ALT_PAB1;; // addr (16-bit)
[j31 + PX_ALT_LOC] = j0;;

// Setup DMA
xr0 = 0x00;;
xr1 = (256 << 16) | 4;;
xr2 = 0x00;;
xr3 = TCB_EPROM | TCB_NORMAL;;

xr4 = 0x00;;
xr5 = (256 << 16) | 1;;
xr6 = 0x00;;
xr7 = TCB_INTMEM | TCB_NORMAL;;

DCS0 = xr3:0;;
DCD0 = xr7:4;;

wait_XPF:
j1:0 = DSTAT;;
j0 = j0 and 0x7;;
comp(j0, DSTAT_DONE);;
if njeq, jump wait_XPF;;

// Jump to target address
cjmp = 0x00;;
cjmp(abs)(np);;

```

Выбор данного варианта соответствует загрузке процессора из внешней параллельной памяти типа EPROM. При выборе режима загрузки EPROM процессор инициализирует канал 0 DMA для передачи 256-ти 32-разрядных слов кода из загрузочного EPROM в ячейки 0x00-0xFF блока 0 внутренней памяти процессора. После инициализации DMA канала, процессор переходит в состояния проверки завершения передачи блока данных из внешней памяти во внутреннюю. Как только передача завершится, процессор передает управление по адресу 0.

### **24.3 Загрузка с использованием порта связи**

Если выбран один из вариантов старта 7,6 или 5, процессор выполняет загрузку начального кода посредством портов связи. При этом различные варианты соответствуют различной ширине шины данных порта связи. Процессор передает управление на код программы, приведенный ниже:

```

start_LINK1:
    jump start_LINK (np);    j0 = LRX_DSIZE_1BIT;;
start_LINK4:
    jump start_LINK (np);    j0 = LRX_DSIZE_4BIT;;
start_LINK8:
    j0 = LRX_DSIZE_8BIT;;

```

```

start_LINK:
    j1 = j0;;
    j0 = j0 or LRX_EN | LRX_CLK_L0;;
    j1 = j1 or LRX_EN | LRX_CLK_L1;;

    // Setup GPIO
    // Alternate function for LINK0 and LINK1 control lines
    j4 = (PF_ALT<<24) | (PF_ALT<<25) | (PF_ALT<<26) | (PF_ALT<<27) | \
        (PF_ALT<<28) | (PF_ALT<<29) | (PF_ALT<<30) | (PF_ALT<<31);;
    [j31 + GPC_ALT_LOC] = j4;;

    // Disable clocks that are not required
    j6 = 0xFFFFFFFF;;
    [j31 + CMU_CLKDIS_LOC] = j6;;

    // Enable LINK clock PLL bypass
    j4 = (1<<PLL_BYPASS_P);;
    [j31 + PLL_LINK_CFG_LOC] = j4;;

    // Reset DMA
    xr4 = 0x00;;
    xr5 = 0x00;;
    xr7:6 = xr5:4;;
    DC8 = xr7:4;;
    DC9 = xr7:4;;

    // Setup DMA
    xr0 = 0x00;;
    xr1 = (256 << 16) | 0x04;;
    xr2 = 0x00;;
    xr3 = TCB_INTMEM | TCB_QUAD;;

    // Enable LINK port receivers
    LRCTL0 = j0;;
    LRCTL1 = j1;;

    // Start DMA
    DC8 = xr3:0;;
    DC9 = xr3:0;;

    // Wait for DMA (any of two channels)
wait_LINK:
    j1:0 = DSTAT;;
    j2 = j1 and (0x7 << 0);;
    j3 = j1 and (0x7 << 3);;
    comp(j2, DSTAT_DONE << 0);;
    if jeq, jump wait_LINK_done(np);;
    comp(j3, DSTAT_DONE << 3);;
    if jeq, jump wait_LINK_done(np);;
    jump wait_LINK;;

wait_LINK_done:

```

```
// Jump to target address
cjmp = 0x00;;
cjmp(abs)(np);;
```

Из текста программы следует, что процессор программирует два канала ДМА для приема данных из портов связи. Каналы DMA портов связи инициализируются для передачи блока 256 слов в адреса внутренней памяти с 0 по 255. После программирования передачи, процессор выполняет ожидание завершения пересылки и затем передает управление по адресу 0.

## 24.4 Старт по запросу прерывания

Если выбран вариант старта 1 или 0, процессор выполняет переход на метку start\_IRQ, с которой начинается выполнение следующего кода программы:

```
start_IRQ:
// Initialize SYSCON register
j0 = j10 or SYSCON_MS0_WT3      |
                                SYSCON_MS0_SLOW      |
                                SYSCON_MS0_IDLE      |
                                SYSCON_MS1_WT3      |
                                SYSCON_MS1_SLOW      |
                                SYSCON_MS1_IDLE;;

syscon = j0;;

// Disable clocks that are not required
j6 = 0xFFFFFFFF;;
[j31 + CMU_CLKDIS_LOC] = j6;;

// Setup GPIO
// Alternate function for nBMS, nRD, MS1 and MS0 signals
j0 = (PF_ALT << 20) | (PF_ALT << 21) | (PF_ALT << 18) | (PF_ALT << 19);;
[j31 + GPC_ALT_LOC] = j0;;

// Alternate function for DATA and ADDR bus
j0 = PX_ALT_PDB0 | PX_ALT_PDB1;; //
data [15:0]
comp(j10, 0x00);;
if njeq; do, j0 = j0 or PX_ALT_PDB2 | PX_ALT_PDB3;; // data
[31:16]
j0 = j0 or PX_ALT_PAB0 | PX_ALT_PAB1 | PX_ALT_PAB2;; // addr
[j31 + PX_ALT_LOC] = j0;;

// Setup vectors
j8 = 0x30000000;;
IVIRQ0 = j8;;
j8 = 0x38000000;;
IVIRQ1 = j8;;
j8 = 0x00000000;;
IVIRQ3 = j8;;
```

```

j8 = 0x0F;;
intctl = j8;; // nIRQ3:0 by level

// Enable interrupts by external IRQs and vector interrupt (for host)
j8 = INT_IRQ0 | INT_IRQ1 /*| INT_IRQ2*/ | INT_IRQ3 | INT_VIRPT;;
IMASKH = j8;;

// SQCTL[GIE] bit is set by reset, so we're done!
stay_idle:
nop;; nop;; nop;; nop;;
idle;; nop;; nop;; nop;;
jump stay_idle(np);;
    
```

В данном варианте старта процессор определяет ширину внешней шины в 32 бита (вариант 1) или 16 бит (вариант 0), конфигурирует контроллер прерываний для обслуживания запросов прерываний по входам IRQ0, IRQ1, IRQ3, предварительно определив для них значения векторов прерываний 0x3000\_0000, 0x3800\_0000 и 0. Также разрешается векторное прерывание. Определив конфигурацию, процессор переходит в режим ожидания прерывания. При возникновении одного из определенных ранее прерываний, процессор перейдет на его обработку. В данном режиме старта, внешний хост через последовательный интерфейс может загрузить код программы во внутреннюю память и, воспользовавшись векторным прерыванием, запустить процессор на выполнение загруженной программы.

## 24.5 Загрузка из внешней последовательной флэш памяти

Если выбран вариант старта номер 3, процессор выполняет переход на метку start\_SPIF, с которой начинается выполнение следующего кода программы:  
start\_SPIF:

```

// Disable clocks that are not required
j6 = ~CMU_CLKDIS_SPI;;
[j31 + CMU_CLKDIS_LOC] = j6;;

// Setup SPI
j0 = (199 << 16) | SPCR0_DSS_8BIT;; // 8-bit mode,
SPI clock = (SCLK_in / 2) / 200
[j31 + SPCR0_LOC] = j0;;
j0 = (1<<SPCR1_HOLDCS_P) | (1<<SPCR1_SPE_P);; // Enable SPI,
hold #CS
[j31 + SPCR1_LOC] = j0;;

// Setup GPIO
j0 = (PF_ALT << 4) | (PF_ALT << 5) | (PF_ALT << 6) | (PF_ALT << 7);;
[j31 + GPA_ALT_LOC] = j0;;

k0 = SPDR_LOC;;
k1 = SPSR_LOC;;
xr9 = 0x0808;; // Initial FDEP arg
xr10 = 0x0800;; // FDEP arg modifier
j10 = 0x00000000;; // target address
j11 = 0x00;;
    
```

```

lc0 = 256;;

// Issue READ command
xr0 = 0x03;;
[k0 + 0] = xr0;          xr1 = 0x00;;
[k0 + 0] = xr1;;      // A[23:16]
[k0 + 0] = xr1;;      // A[15:8]
[k0 + 0] = xr1;;      // A[7:0]

loop_SPIF:
// Write 4 bytes
[k0 + 0] = xr1;;

// Wait 4 bytes to be read
wait_SPIF:
j0 = [k1 + 0];;
j0 = j0 and (1<<SPSR_RFS_P);;
if jeq, jump wait_SPIF;;

// Read received data and compose 32-bit word
xr8 = xr9;;
xr4 = [k0 + 0];;
xr0 = [k0 + 0];;
xr4 += fdep r0 by r8;;
xr0 = [k0 + 0];          xr8 = r8 + r10;;
xr4 += fdep r0 by r8;;
xr0 = [k0 + 0];          xr8 = r8 + r10;;
xr4 += fdep r0 by r8;;

// First read is dummy due to command and address shift
[j10 += j11] = xr4;;
if nlc0e, jump loop_SPIF; j11 = 0x01;;

// There are 4 more bytes in the FIFO - read them
comp(j10, 0xFF);;
if jeq, jump wait_SPIF(np);lc0 = 0x01;;

// Done! Release #CS0
j0 = /*(1<<SPCR1_HOLDCS_P) */ (1<<SPCR1_SPE_P);;
[j31 + SPCR1_LOC] = j0;;

// Jump to target address
cjmp = 0x00000000;;
cjmp(abs)(np);;

```

## 25 Часы реального времени RTC

Модуль RTC представляет собой набор счетчиков, которые используют для своей работы входную частоту процессора ХТІ. В связи с отсутствием частоты 32,768 КГц обычно используемой для часов реального времени, в данном модуле используется предварительный делитель входной частоты ХТІ с помощью которого можно задать опорную частоту, выполняющую такие же функции, что и частота 32,768 КГц.

Модуль RTC включает набор регистров, позволяющих управлять работой модуля, а также отслеживать его состояние. Регистры подключены к шине обмена периферийных устройств и имеют базовый адрес **0x800001E0**. Краткое описание регистров приведено ниже (Таблица 182).

**Таблица 182 – Регистры модуля RTC**

Номер	Название	Бит	Сброс	Регистр
0	RTC_CNT	32	0	Счетчик секунд
1	RTC_MR	32	max	Регистр сравнения
2	WDT_CNT	8	15	Текущее значение сторожевого счетчика
3	RTC_TDIV	20	0	Текущее значение делителя тик-импульса
4	RTC_CR	8	0	Регистр управления
5	RTC_SDIV	16	0	Текущее значение делителя секунды
6	TIC_VAL	20	255	Период тик-импульса минус 1
7	SEC_VAL	16	3	Количество тик-импульсов в секунде
8	RTC_BUSY	1	0	Флаг занятости интерфейса

Счетчики модуля RTC также могут быть использованы как дополнительные таймерные модуля, используемые для генерации различных временных интервалов.

### 25.1 Формирователь «тик-импульсов»

Входная частота ХТІ поступает на счетчик RTC\_TDIV, который по каждому импульсу входного синхросигнала увеличивает свое значение на 1. Значение счетчика RTC\_TDIV непрерывно сравнивается со значением регистра TIC\_VAL. Если значение счетчика RTC\_TDIV становится равным значению регистра TIC\_VAL, то с приходом следующего импульса ХТІ значение счетчика RTC\_TDIV становится равным нулю и к контроллеру прерывания посылается запрос прерывания – «тик-импульс». Регистр TIC\_VAL задает период следования тик-импульсов и содержит значение реального периода уменьшенное на 1. Тик-импульсы могут быть использованы операционной системой в качестве базового периода для работы программ в системе разделения времени. Период формирования тик-импульсов равен

$$T_{\text{тик}} = XTI / (TIC\_VAL + 1).$$

Полученные тик-импульсы используются модулем RTC для генерации импульсов секунд (точнее – псевдосекунд) и для работы сторожевого таймера.

## **25.2 Формирователь импульсов секунд**

Модуль RTC содержит счетчик RTC\_SDIV, используемый для формирования импульсов секунд. Счетчик тактируется с помощью синхросигнала ХТІ, однако увеличивает значение RTC\_SDIV на 1, только в момент прихода тик-импульса. Значение регистра RTC\_SDIV постоянно сравнивается со значением регистра SEC\_VAL, который задает период работы счетчика RTC\_SDIV в терминах тик-импульсов. Если значение счетчика RTC\_SDIV равно значению регистра SEC\_VAL, то с приходом следующего тик-импульса будет сформирован импульс секунды. Значение счетчика RTC\_SDIV станет равным нулю, и он начнет повторный счет. Таким образом генератор секунд формирует импульсы секунд согласно формуле

$$T_{\text{сек}} = T_{\text{тик}} / (\text{SEC\_VAL} + 1).$$

Секундные импульсы поступают на счетчик секунд RTC\_CNT.

## **25.3 Счетчик секунд RTC\_CNT.**

В данном счетчике содержится время в секундах прошедшее с момента включения часов. Счетчик может быть скорректирован. Счетчик тактируется синхросигналом ХТІ, однако увеличивает свое значение на 1 только в момент прихода импульса секунд. Регистр RTC\_CNT имеет 32 разряда. Его значение непрерывно сравнивается со значением регистра RTC\_MR. В случае если значение счетчика RTC\_CNT равно значению регистра RTC\_MR, то в момент прихода следующего секундного импульса будет сформирован запрос прерывания ALARM к процессору. Регистр RTC\_MR позволяет реализовать функции будильника и должен содержать значение времени срабатывания в секундах (минус 1 от действительного значения).

### **25.3.1 Регистр сравнения RTC\_MR**

Регистр сравнения используется для реализации функций будильника. Его значение непрерывно сравнивается со значением регистра счетчика секунд. В момент равенства значений устанавливается соответствующий флаг и вырабатывается сигнал прерывания.

### **25.3.2 Регистр управления RTC\_CR**

Регистр управляет включением или выключением различных функций часов реального времени. Биты регистра кратко описаны ниже (). По сигналу внешнего сброса все биты регистра устанавливаются в ноль.

**Таблица 183 – Регистр управления**

<b>Бит</b>	<b>Имя</b>	<b>Тип</b>	<b>Назначение</b>
1:0			
2	WDT_EN	R/W	Разрешение работы сторожевого таймера 0 – выключен 1 – включен
3	FREEZ	R/W	Блокировка сторожевого таймера. После установки этого

			бита невозможно выключить сторожевой таймер.
6:4	WDT_SEL	R/W	Выбор значения загружаемого в счетчик сторожевого таймера. 0 – 1, 1 – 2, 2 – 4, 3 – 8, 4 – 16, 5 – 32, 6 – 64, 7 – 128
7	LOCK	R/W	Управление доступом к модулю RTC 0 – чтение-запись 1 – только чтение (для RTC_CR – чтение-запись)
8	DIS_ATIC	R/W	Когда 1 запрещает генерацию асинхронного прерывания от импульса «тик»
9	DIS_ADOG	R/W	Когда 1 запрещает генерацию асинхронного прерывания от сторожевого таймера
10	DIS_AMR	R/W	Когда 1 запрещает генерацию асинхронного прерывания от регистра сравнения

Функции регистра управления заключаются в задании параметров работы модуля RTC. Биты регистра управления позволяют включить сторожевой таймер (WDT\_EN), при необходимости заблокировать возможность его случайного выключения(FREEZ). С помощью разрядов WDT\_SEL можно задать значение константы, которое будет загружаться в сторожевой таймер при его инициализации. Также возможно задание блокировки доступа к регистрам RTC для защиты от случайной записи (бит LOCK). В этом случае ко всем регистрам RTC (кроме регистра управления) возможен доступ только по чтению. В процессоре реализован режим пониженного энергопотребления (спящий режим), в котором все системные частоты могут быть остановлены. Активной может быть только частота ХТІ. В этом случае для выхода из спящего режима могут быть использованы асинхронные запросы прерываний от тик-импульса, будильника или сторожевого таймера. Соответствующие биты регистра управления разрешают эти функции.

### **25.3.3 Регистр счетчика сторожевого таймера WDT\_CNT**

Этот 8-разрядный регистр осуществляет вычитание 1 из своего содержимого при поступлении тик-импульса. При достижении значения WDT\_CNT==1 и в момент прихода следующего тик-импульса, сторожевой таймер вырабатывает запрос на прерывание. При этом значение WDT\_CNT становится равным нулю. Запрос прерывания от сторожевого таймера должен рассматриваться как запрос на повторную инициализацию сторожевого таймера. Если значение WDT\_CNT равно нулю и приходит очередной тик-импульс, сторожевой таймер сформирует сигнал сброса системы длительностью в один период тик-импульса. Косвенным образом можно записать новое значение в счетчик сторожевого таймера. Для этого вначале нужно записать соответствующее значение в биты WDT\_SEL регистра управления, а затем записать константу 0x04072013 по адресу счетчика сторожевого таймера. В регистре WDT\_CNT автоматически установится требуемое значение.

### **25.3.4 Регистр занятости интерфейса часов RTC\_BUSY**

Это 1-битный регистр, доступный только по чтению. Проблема в работе с модулем RTC заключается в том, что все его внутренние регистры тактируются частотой ХТІ. В тоже время процессор обращается к модулю через периферийную шину, частота которой может значительно превосходить частоту ХТІ. В связи с этим в модуле реализован согласующий буфер между двумя доменами синхронизации. Любая запись в часы приводит к записи в буфер. Значение из буфера в дальнейшем передается в другой домен и там происходит запись. Пока вся процедура не выполнена, флаг занятости равен 1 и новая запись невозможна. А именно новая запись приведет к ошибке. Чтение регистров реализовано следующим образом. При чтении регистра модуля часов происходит копирование значения регистра в буфер

периферийной шины. В этом же такте старое значение буфера передается в процессор. Таким образом, чтобы прочитать текущее значение какого-то регистра необходимо выполнить два последовательных чтения.

Отметим, что при чтении регистра **RTC\_BUSY** считывается непосредственное значение флага, и нет необходимости выполнять чтение повторно.

## 25.4 Ограничения модуля RTC

Модуль использует для своей работы входную тактовую частоту ХТІ. При генерации запросов прерывания от тик-импульсов, будильника, сторожевого таймера, импульсы запросов синхронизируются частотой SOC-шины. Для надежной генерации прерывания необходимо чтобы частота SOC-шины была выше внешней входной частоты ХТІ.

## 25.5 Состояние модуля RTC после сброса

Процессор может быть сброшен в исходное состояние следующими сигналами:

- внутренний сигнал сброса от модуля контроля включения питания процессора nPOR;
- активным уровнем по внешнему входу nRST\_in;
- сигналом от сторожевого таймера WDT\_RES.

Часть ресурсов модуля RTC сбрасывается сигналом nPOR только при включении питания процессора. Этими ресурсами являются регистры: RTC\_CNT, RTC\_MR, RTC\_TDIV, RTC\_SDIV, TIC\_VAL, SEC\_VAL. Все другие регистры и флаги модуля сбрасываются любым из трех вышеназванных источников. Таким образом, если пользователь определил конфигурацию часов реального времени, значение часов будет корректным до тех пор пока питание процессора в норме. Никакие внешние события или внутренний сброс от сторожевого таймера не изменяют значение часов. В модуле CMU имеется флаг, который позволяет различить сброс сторожевого таймера от сброса из других источников. Однако нет возможности различить сброс по включению питания от внешнего сброса. Между тем факт внешнего сброса может быть вычислен косвенным образом. Например, если после сброса значение счетчика секунд не равно нулю, значит произошел сброс от внешнего сигнала и программа не должна выполнять процедуру настройки часов повторно.

Регистр управления модулем RTC\_CR сбрасывается в любом из трех случаев.

## 26 Пределно-допустимые режимы

Таблица 184 – Пределно-допустимые режимы эксплуатации и предельные электрические режимы микросхем

Наименование параметра, единица измерения	Буквенное обозначение параметра	Норма параметра			
		Пределно- допустимый режим		Пределный режим	
		не менее	не более	не менее	не более
Напряжение питания ядра, В	U <sub>CC</sub>	1,08	1,32	–	1,4
Напряжение питания ввода/вывода, В	U <sub>CCIO</sub>	3,0	3,6	–	4,0
Напряжение питания аналоговых блоков, В	U <sub>CCA</sub>	3,0	3,6	–	4,0
Входное напряжение высокого уровня, В	U <sub>IH</sub>	2	U <sub>CCIO</sub>	U <sub>CCIO</sub>	U <sub>CCIO</sub> +0,3
Входное напряжения низкого уровня, В на всех выводах кроме ХТ1	U <sub>IL</sub>	0	0,7	– 0,3	0
на выводе ХТ1			0,4		
Входное напряжение дифференциальное, В	U <sub>ID</sub>	0,2	1,2	–	–
Входное напряжение синфазное, В	U <sub>IS</sub>	0,6	1,8	–	–
Выходной ток цифровых выходов высокого уровня, мА	I <sub>OH</sub>	– 8*	0	– 24	–
Выходной ток цифровых выходов низкого уровня, мА	I <sub>OL</sub>	0	8	–	24
Тактовая частота процессора, МГц	f <sub>C</sub>	–	300	–	–
Входная частота ФАПЧ, МГц	f <sub>C_PLL</sub>	10	100	–	–
Тактовая частота Link порта, МГц	f <sub>C_LINK</sub>	–	300	–	–
Тактовая частота JTAG, МГц	f <sub>JTAG</sub>	1	10	–	–
<p>*Суммарный выходной ток высокого уровня не должен быть более 800 мА.  <i>Примечание</i> – Не допускается одновременное воздействие двух и более предельных режимов</p>					

## 27 Электрические параметры

Таблица 185 – Электрические параметры микросхем при приемке и поставке

Наименование параметра, единица измерения, режим измерения	Буквенное обозначение параметра	Норма параметра		Температура среды, <sup>1)</sup> °С
		не менее	не более	
Выходное напряжение высокого уровня, В	U <sub>OH</sub>	2,0	–	25, 105, – 60
Выходное напряжение низкого уровня, В	U <sub>OL</sub>	–	0,4	25, 105, – 60
Дифференциальное выходное напряжение Link портов уровней логического «0» и логической «1», мВ	U <sub>OD</sub>	150	650	25, 105, – 60
Синфазное выходное напряжение Link портов уровней логического «0» и логической «1», В	U <sub>OS</sub>	1,10	1,55	25, 105, – 60
Ток утечки высокого уровня на входе, мкА	I <sub>I<sub>LH</sub></sub>	–20	20	25, 105, – 60
Ток утечки низкого уровня на входе, мкА	I <sub>I<sub>LL</sub></sub>	– 20	20	25, 105, – 60
Входной ток высокого уровня на выводе с внутренним резистором доопределения до нуля, мкА	I <sub>I<sub>H_D</sub></sub>	–	800	25, 105, – 60
Входной ток низкого уровня на выводе с внутренним резистором доопределения до питания, мкА	I <sub>I<sub>L_U</sub></sub>	– 800	–	25, 105, – 60
Динамический ток потребления схем ввода-вывода, мА	I <sub>o<sub>CC</sub></sub>	–	800	25, 105, – 60
Потребляемая мощность, Вт	P <sub>CC</sub>	–	4	25, 105, – 60
Выходная частота блока PLL (измеряется после делителя), МГц	f <sub>V<sub>CO</sub></sub>	800	1600	25, 105, – 60

## 28 Габаритный чертеж микросхемы

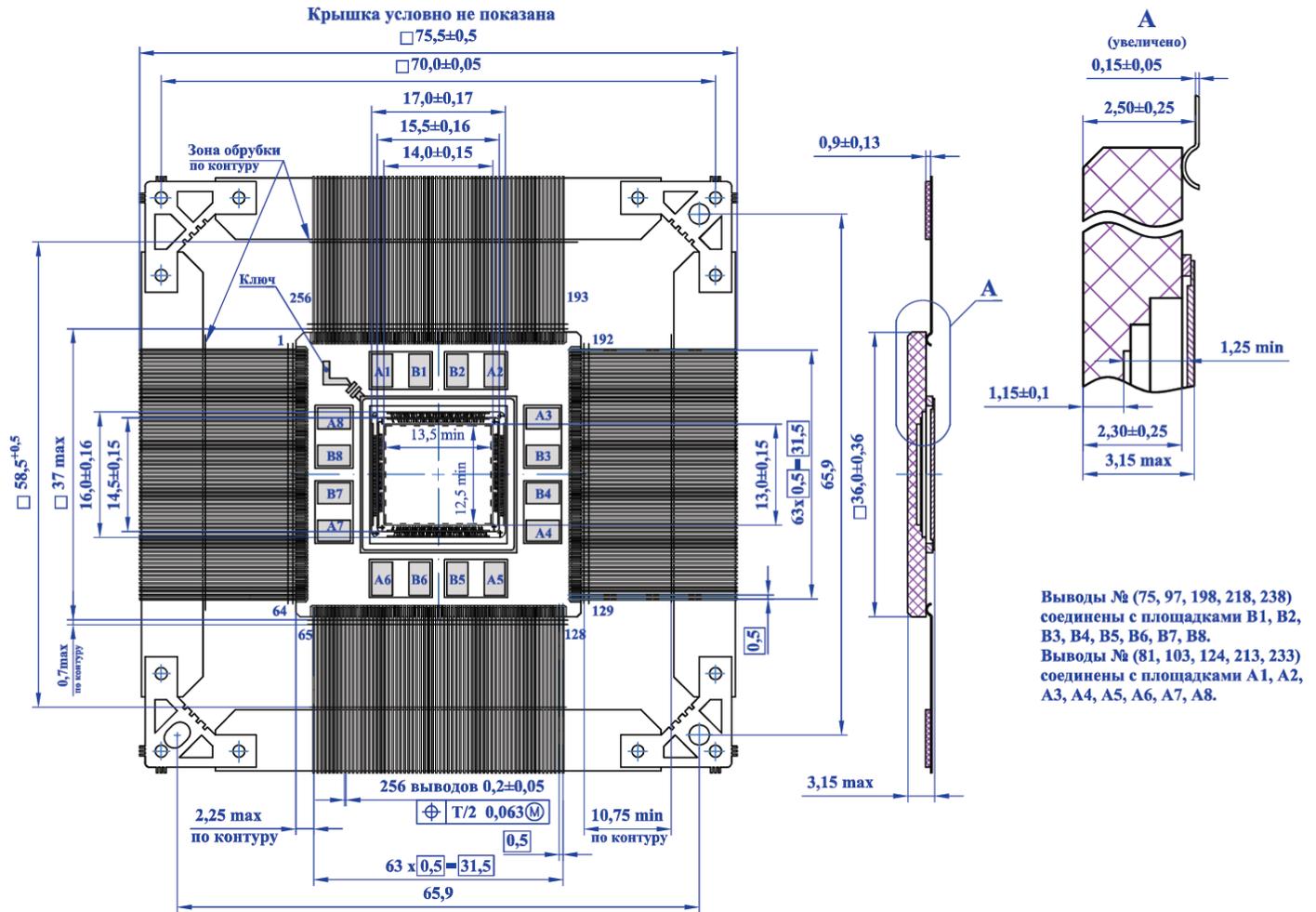


Рисунок 76 – Микросхема в корпусе 4244.256-3

## 29 Информация для заказа

Обозначение	Маркировка	Тип корпуса	Температурный диапазон
1967ВЦ3Т	1967ВЦ3Т	4244.256-3	минус 60 – 125 °С
К1967ВЦ3Т	К1967ВЦ3Т	4244.256-3	минус 60 – 125 °С
К1967ВЦ3ТК	К1967ВЦ3Т●	4244.256-3	0 – 70 °С

Микросхемы с приемкой «ВП» маркируются ромбом.

Микросхемы с приемкой «ОТК» маркируются буквой «К».

## Лист регистрации изменений

№ п/п	Дата	Версия	Краткое содержание изменения	№№ изменяемых листов
1	22.01.2015	0.1.0	Введена вновь	–
2	06.02.2015	0.1.1	Исправлена маркировка даты на рисунке	1
3	31.08.2015	0.2.0	Корректировка по замечаниям и дополнениям разработчика	Все
4	02.10.2015	0.3.0	Исправления в таблице 1	20
5	23.10.2015	0.4.0	Исправлено значение напряжения питания	1, 316
6				
7				
8				
9				
10				